8th Innovations in Theoretical Computer Science Conference

ITCS 2017, January 9–11, 2017 - Berkeley, CA, USA

Edited by

Christos H. Papadimitriou

Editors

Christos H. Papadimitriou
Columbia University, New York City
christos@columbia.edu

ACM Classification 1998
F. Theory of Computation, G. Mathematics of Computing

ISBN 978-3-95977-029-3

Published online and open access by
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at http://www.dagstuhl.de/dagpub/978-3-95977-029-3.

Publication date
November, 2017

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

License
This work is licensed under a Creative Commons Attribution 3.0 Unported license (CC-BY 3.0): http://creativecommons.org/licenses/by/3.0/legalcode.

In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:

- Attribution: The work must be attributed to its authors.

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/LIPIcs.ITCS.2017.0

LIPIcs – Leibniz International Proceedings in Informatics

LIPIcs is a series of high-quality conference proceedings across all fields in informatics. LIPIcs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

Editorial Board

- Luca Aceto (*Chair*, Gran Sasso Science Institute and Reykjavik University)
- Susanne Albers (TU München)
- Chris Hankin (Imperial College London)
- Deepak Kapur (University of New Mexico)
- Michael Mitzenmacher (Harvard University)
- Madhavan Mukund (Chennai Mathematical Institute)
- Anca Muscholl (University Bordeaux)
- Catuscia Palamidessi (INRIA)
- Raimund Seidel (Saarland University and Schloss Dagstuhl – Leibniz-Zentrum für Informatik)
- Thomas Schwentick (TU Dortmund)
- Reinhard Wilhelm (Saarland University)

ISSN 1868-8969

http://www.dagstuhl.de/lipics

Contents

Papers

8th Innovations in Theoretical Computer Science Conference (ITCS 2017).
Editor: Christos H. Papadimitrou
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Preface

The 8th Innovations in Theoretical Computer Science (ITCS) Conference was held between the 9th and 11th of January 2017 at the University of California at Berkeley, sponsored by the Simons Institute for the Theory of Computing. The role of ITCS is to complement the two major, prestigious, and very successful annual conferences of our field, STOC and FOCS by enabling a dialogue that is crucial and beneficial for the unity, vigor, and future of our field, and providing a forum for ideas that are novel and forward looking. ITCS conferences are typically small (about a hundred participants) with a well attended single track.

171 papers were submitted to the 8th ITCS, and from these the program committee selected for presentation at the conference and publication in these proceedings a total of 61 papers. Six papers were *invited,* meaning that they were of such quality that the program committee either explicitly invited their authors to submit them, or realized in retrospect that they should have done so: "Separators in region intersection graphs" by James R. Lee; "IRLS and Slime Mold" by Damian Straszak and Nisheeth Vishnoi; "Network coding in undirected graphs is either very helpful or not helpful at all" by Mark Braverman, Sumegha Garg and Ariel Schvartzman; "Nash Social Welfare, Matrix Permanent, and Stable Polynomials" by Nima Anari, Shayan Oveis Gharan, Amin Saberi and Mohit Singh; "Multiplayer parallel repetition for expander games" by Irit Dinur, Prahladh Harsha, Rakesh Venkat and Henry Yuen; and "The Journey from NP to TFNP Hardness" by Pavel Hubacek, Moni Naor and Eylon Yogev. Other than that, there is no best paper award at ITCS. The best student paper award was shared by these two papers: "Towards Hardness of Approximation for Polynomial Time Problems" by Amir Abboud and Arturs Backurs and "Detecting communities is hard and counting them is even harder," by Aviad Rubinstein.

Many heartfelt thanks are due to the stellar and hard working program committee, consisting of: Scott Aaronson, Elette Boyle, Mark Braverman, Alessandro Chiesa, Artur Czumaj, Costis Daskalakis, Shafi Goldwasser, Anna Karlin, Jon Kleinberg, Swastik Kopparty, Muthu Muthukrishnan, Noam Nisan, Georgios Piliouras, Toniann Pitassi, Tal Rabin, Alexander Razborov, Tim Roughgarden, Aviad Rubinstein, Nikhil Srivastava, Chris Umans, Paul Valiant, Virginia Vassilevska-Williams, Umesh Vazirani, Santosh Vempala, Mary Wootters, Nir Yosef, Henry Yuen, and Lisa Zhang. Of these, Ale Chiesa and Umesh Vazirani were also local organizers, and they worked very hard for the success of the conference. We are most grateful to the Simons Institute for hosting and supporting the conference. Finally, many thanks to all authors who submitted their work, and especially to all participants for making this a truly memorable event.

8th Innovations in Theoretical Computer Science Conference (ITCS 2017).
Editor: Christos H. Papadimitrou
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Separators in Region Intersection Graphs

James R. Lee

Computer Science, University of Washington, Seattle, USA
jrl@cs.washington.edu

Abstract

For undirected graphs $G = (V, E)$ and $G_0 = (V_0, E_0)$, say that G is a *region intersection graph over* G_0 if there is a family of connected subsets $\{R_u \subseteq V_0 : u \in V\}$ of G_0 such that $\{u, v\} \in E \iff R_u \cap R_v \neq \emptyset$.

We show if G_0 excludes the complete graph K_h as a minor for some $h \geq 1$, then every region intersection graph G over G_0 with m edges has a balanced separator with at most $c_h\sqrt{m}$ nodes, where c_h is a constant depending only on h. If G additionally has uniformly bounded vertex degrees, then such a separator is found by spectral partitioning.

A string graph is the intersection graph of continuous arcs in the plane. String graphs are precisely region intersection graphs over planar graphs. Thus the preceding result implies that every string graph with m edges has a balanced separator of size $O(\sqrt{m})$. This bound is optimal, as it generalizes the planar separator theorem. It confirms a conjecture of Fox and Pach (2010), and improves over the $O(\sqrt{m}\log m)$ bound of Matoušek (2013).

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.1.6 Optimization, G.2.2 Graph Theory

Keywords and phrases Graph separators, planar graphs, spectral partitioning

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.1

1 Introduction

Consider an undirected graph $G_0 = (V_0, E_0)$. A graph $G = (V, E)$ is said to be a *region intersection graph (rig) over* G_0 if the vertices of G correspond to connected subsets of G_0 and there is an edge between two vertices of G precisely when those subsets intersect. Concretely, there is a family of connected subsets $\{R_u \subseteq V_0 : u \in V\}$ such that $\{u, v\} \in E \iff R_u \cap R_v \neq \emptyset$. For succinctness, we will often refer to G as a *rig over* G_0.

Let $\mathsf{rig}(G_0)$ denote the family of all finite rigs over G_0. Prominent examples of such graphs include the intersection graphs of pathwise-connected regions on a surface (which are intersection graphs over graphs that can be drawn on that surface).

For instance, *string graphs* are the intersection graphs of continuous arcs in the plane. It is easy to see that every finite string graph G is a rig over some planar graph: By a simple compactness argument, we may assume that every two strings intersect a finite number of times. Now consider the planar graph G_0 whose vertices lie at the intersection points of strings and with edges between two vertices that are adjacent on a string (see Figure 1). Then $G \in \mathsf{rig}(G_0)$. It is not too difficult to see that the converse is also true; see Section 4.

To illustrate the non-trivial nature of such objects, we recall that there are string graphs on n strings that require $2^{\Omega(n)}$ intersections in any such representation [14]. The recognition problem for string graphs is NP-hard [13]. Decidability of the recognition problem was established in [25], and membership in NP was proved in [24]. We refer to the recent survey [22] for more of the background and history behind string graphs.

Even when G_0 is planar, the rigs over G_0 can be dense: Every complete graph is a rig over some planar graph (in particular, every complete graph is a string graph). It has been


© James R. Lee;
licensed under Creative Commons License CC-BY

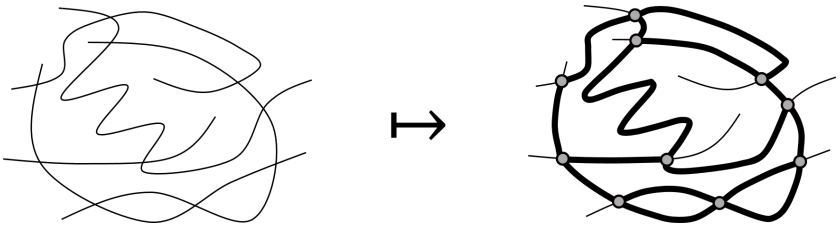


Figure 1 A string graph as a rig over a planar graph.

conjectured by Fox and Pach [5] that every m-edge string graph has a balanced separator with $O(\sqrt{m})$ nodes. Fox and Pach proved that such graphs have separators of size $O(m^{3/4}\sqrt{\log m})$ and presented a number of applications of their separator theorem. Matoušek [21] obtained a near-optimal bound of $O(\sqrt{m}\log m)$. In the present work, we confirm the conjecture of Fox and Pach, and generalize the result to include all rigs over graphs that exclude a fixed minor. This extended abstract contains mostly theorem statements; for detailed proofs and further arguments, we refer to the full paper [15].

▶ **Theorem 1.** *If $G \in \mathsf{rig}(G_0)$ and G_0 excludes K_h as a minor, then G has a $\frac{2}{3}$-balanced separator of size at most $c_h\sqrt{m}$ where m is the number of edges in G. Moreover, one has the estimate $c_h \leq O(h^3\sqrt{\log h})$.*

In the preceding statement, an ϵ-balanced separator of $G = (V, E)$ is a subset $S \subseteq V$ such that in the induced graph $G[V \setminus S]$, every connected component contains at most $\epsilon|V|$ vertices.

The proof of Theorem 1 is constructive, as it is based on solving and rounding a linear program; it yields a polynomial-time algorithm for constructing the claimed separator. In the case when there is a bound on the maximum degree of G, one can use the well-known spectral bisection algorithm (see Section 1.5).

Since planar graphs exclude K_5 as a minor, Theorem 1 implies that m-edge string graphs have $O(\sqrt{m})$-node balanced separators. Since the graphs that can be drawn on any compact surface of genus g exclude a K_h minor for $h \leq O(\sqrt{g+1})$, Theorem 1 also applies to string graphs over any fixed compact surface.

In addition, it implies the Alon-Seymour-Thomas [1] separator theorem[1] for graphs excluding a fixed minor, for the following reason. Let us define the *subdivision of a graph G* to be the graph $\dot{G}$ obtained by subdividing every edge of G into a path of length two. Then every graph G is a rig over $\dot{G}$, and it is not hard to see that for $h \geq 1$, G has a K_h minor if and only if $\dot{G}$ has a K_h minor.

1.1 Applications in topological graph theory

We mention two applications of Theorem 1 in graph theory. In [6], the authors present some applications of separator theorems for string graphs. In two cases, the tight bound for separators leads to tight bounds for other problems. The next two theorems confirm conjectures of Fox and Pach; as proved in [6], they follow from Theorem 1. Both results are tight up to a constant factor.

[1] Note that Theorem 1 is quantitatively weaker in the sense that [1] shows the existence of separators with $O(h^{3/2}\sqrt{n})$ vertices. Since every K_h-minor-free graph has at most $O(nh\sqrt{\log h})$ edges [12, 27], our bound is $O(h^{7/2}(\log h)^{3/4}\sqrt{n})$.

▶ **Theorem 2.** *There is a constant $c > 0$ such that for every $t \geq 1$, it holds that every $K_{t,t}$-free string graph on n vertices has at most $cnt(\log t)$ edges.*

A *topological graph* is a graph drawn in the plane so that its vertices are represented by points and its edges by curves connecting the corresponding pairs of points.

▶ **Theorem 3.** *In every topological graph with n vertices and $m \geq 4n$ edges, there are two disjoint sets, each of cardinality*

$$\Omega\left(\frac{m^2}{n^2 \log \frac{n}{m}}\right) \tag{1}$$

so that every edge in one set crosses all edges in the other.

This improves over the bound of $\Omega\left(\frac{m^2}{n^2(\log \frac{n}{m})^c}\right)$ for some $c > 0$ proved in [7], where the authors also show that the bound (1) is tight. Before we conclude this section, let us justify the observation made earlier.

▶ **Lemma 4.** *Finite string graphs are precisely finite region intersection graphs over planar graphs.*

Proof. We have already argued that string graphs are planar rigs. Consider now a planar graph $G_0 = (V_0, E_0)$ and a finite graph $G = (V, E)$ such that $G \in \mathsf{rig}(G_0)$. Let $\{R_u \subseteq V_0 : u \in V\}$ be a representation of G as a rig over G_0.

Since G is finite, we may assume that each region R_u is finite. To see this, for $v \in V_0$, let its *type* be the set $T(v) = \{u \in V : v \in R_u\}$. Then since G is finite, there are only finitely many types. For any region $R_u \subseteq V_0$, let $\tilde{R}_u$ be a finite set of vertices that exhausts every type in R_u, and let $\hat{R}_u$ be a finite spanning tree of $\tilde{R}_u$ in the induced graph $G_0[R_u]$. Then the regions $\{\hat{R}_u : u \in V\}$ are finite and connected, and also form a representation of G as a rig over G_0.

When each region R_u is finite, we may assume also that G_0 is finite. Now take a planar drawing of G_0 in $\mathbb{R}^2$ where the edges of G_0 are drawn as continuous arcs, and for every $u \in V$, let $T_u \subseteq \mathbb{R}^2$ be the drawing of the spanning tree of R_u. Each T_u can be represented by a string (simply trace the tree using an in-order traversal that begins and ends at some fixed node), and thus G is a string graph. ◀

1.2 Balanced separators and extremal spread

Since complete graphs are string graphs, we do not have access to topological methods based on the exclusion of minors. Instead, we highlight a more delicate structural theory. The following fact is an exercise.

Fact: If $\dot{G}$ is a string graph, then G is planar.

More generally, we recall that *H is a minor of G* if H can be obtained from G by a sequence of edge contractions, edge deletions, and vertex deletions. If H can be obtained using only edge contractions and vertex deletions, we say that *H is a strict minor of G.*

▶ **Lemma 5.** *If $G \in \mathsf{rig}(G_0)$ and $\dot{H}$ is a strict minor of G, then H is a minor of G_0.*

This topological structure of (forbidden) strict minors in G interacts nicely with "conformal geometry" on G. Consider the family of all pseudo-metric spaces that arise from a finite graph G by assigning non-negative lengths to its edges and taking the induced shortest path

distance. Certainly if we add an edge to G, the family of such spaces can only grow (since by giving the edge length equal to the diameter of the space, we effectively remove it from consideration). In particular, if $G = K_n$ is the complete graph on n vertices, then every n-point metric space is a path metric on G.

A significant tool will be the study of extremal conformal metrics on a graph G. Unlike in the edge-weighted case, the family of path distances coming from conformal metrics can be well-behaved even if G contains arbitrarily large complete graph minors. As a simple example, let $K_{\mathbb{N}}$ denote the complete graph on countably many vertices. Then every distance arising from a conformal metric on $K_{\mathbb{N}}$ is bi-Lipschitz to an ultrametric.

1.3 Vertex expansion and observable spread

Fix a graph $G = (V_G, E_G) \in \mathsf{rig}(G_0)$ with $n = |V_G|$ and $m = |E_G|$. Since the family $\mathsf{rig}(G_0)$ is closed under taking induced subgraphs, a standard reduction allows us to focus on finding a subset $U \subseteq V_G$ with small isoperimetric ratio: $\frac{|\partial U|}{|U|} \lesssim \frac{\sqrt{m}}{n}$, where

$$\partial U = \{v \in U : E_G(v, V_G \setminus U) \neq \emptyset\},$$

and $E_G(v, V_G \setminus U)$ is the set of edges between v and vertices outside U. Also define the interior $U^\circ = U \setminus \partial U$.

Let us define the *vertex expansion constant* of G as

$$\phi_G = \min\left\{\frac{|\partial U|}{|U|} : \emptyset \neq U \subseteq V_G, |U^\circ| \leq \frac{|V_G|}{2}\right\}. \tag{2}$$

In [4], it is shown that this quantity is related to the concentration function (in the sense of Lévy and Milman; see also Gromov's observable diameter [8]) of extremal conformal metrics on G .

For a finite metric space (X, dist), we define the *spread of* X as the quantity

$$\mathfrak{s}(X, \mathrm{dist}) = \frac{1}{|X|^2}\sum_{x,y\in X} \mathrm{dist}(x, y).$$

Define the *observable spread of* X by

$$\mathfrak{s}_{\mathrm{obs}}(X, \mathrm{dist}) = \sup_{f:X\to\mathbb{R}}\left\{\frac{1}{|X|^2}\sum_{x,y\in X}|f(x) - f(y)| : f \text{ is 1-Lipschitz}\right\}. \tag{3}$$

▶ Remark. We remark on the terminology: In general, it is difficult to "view" a large metric space all at once; this holds both conceptually and from an algorithmic standpoint. If one thinks of Lipschitz maps $f : X \to \mathbb{R}$ as "observations" then the observable spread captures how much of the spread can be "seen."

We then define the L^1*-extremal observable spread of* G as

$$\bar{\mathfrak{s}}_{\mathrm{obs}}(G) = \sup_{\omega:V_G\to\mathbb{R}_+}\left\{\mathfrak{s}_{\mathrm{obs}}(V_G, \mathrm{dist}_\omega) : \|\omega\|_{L^1(V_G)} \leq 1\right\}, \tag{4}$$

where $\|\omega\|_{L^1(V_G)} \coloneqq \frac{1}{|V_G|}\sum_{v\in V_G}\omega(v)$. We recall the following theorem from [4] that relates expansion to the observable spread.

▶ **Theorem 6** ([4]). *For every finite graph* G*,*

$$\tfrac{1}{2}\bar{\mathfrak{s}}_{\mathrm{obs}}(G) \leq \frac{1}{\phi_G} \leq 3\bar{\mathfrak{s}}_{\mathrm{obs}}(G).$$

▶ **Example 7.** If G is the subgraph of the lattice $\mathbb{Z}^d$ on the vertex set $\{0, 1, \ldots, L\}^d$, then $\phi_G \asymp 1/L$ and $\bar{\mathfrak{s}}(G) \asymp L$. This can be achieved by taking $\omega \equiv 1$ and defining $f : V_G \to \mathbb{R}$ by $f(x) = x_1$.

In light of Theorem 6, to prove Theorem 1, it suffices to give a lower bound on $\bar{\mathfrak{s}}_{\mathrm{obs}}(G)$. It is natural to compare this quantity to the *L^1-extremal spread of G*:

$$\bar{\mathfrak{s}}(G) \coloneqq \max \left\{ \frac{1}{|V_G|^2} \sum_{u,v \in V_G} \mathrm{dist}_\omega(u, v) : \|\omega\|_{L^1(V_G)} \le 1 \right\}. \tag{5}$$

Let us examine these two notions for planar graphs using the theory of circle packings.

▶ **Example 8** (Circle packings)**.** Suppose that G is a finite planar graph. The Koebe-Andreev-Thurston circle packing theorem asserts that G is the tangency graph of a family $\{D_v : v \in V_G\}$ of circles on the unit sphere $\mathbb{S}^2 \subseteq \mathbb{R}^3$. Let $\{c_v : v \in V_G\} \subseteq \mathbb{S}^2$ and $\{r_v > 0 : v \in V_G\}$ be the centers and radii of the circles, respectively. An argument of Spielman and Teng [26] (see also Hersch [9] for the analogous result for conformal mappings) shows that one can take $\sum_{v \in V_G} c_v = \mathbf{0}$.

If we define $\omega(v) = r_v$ for $v \in V_G$, then $\mathrm{dist}_\omega \ge \mathrm{dist}_{\mathbb{S}^2} \ge \mathrm{dist}_{\mathbb{R}^3}$ on the centers $\{c_v : v \in V_G\}$. (The latter two distances are the geodesic distance on $\mathbb{S}^2$ and the Euclidean distance on $\mathbb{R}^3$, respectively).

Using the fact that $\sum_{v \in V_G} c_v = \mathbf{0}$, we have

$$\sum_{u,v \in V_G} \|c_u - c_v\|_2^2 = 2n \sum_{u \in V_G} \|c_v\|^2 = 2n^2. \tag{6}$$

This yields

$$\sum_{u,v \in V_G} \mathrm{dist}_\omega(u, v) \ge \sum_{u,v \in V_G} \|c_u - c_v\| \ge \frac{n^2}{2}.$$

Moreover,

$$\|\omega\|_{L^1(V_G)} \le \|\omega\|_{L^2(V_G)} = \sqrt{\frac{1}{n} \sum_{v \in V_G} r_v^2} \le \sqrt{\frac{\mathrm{vol}(\mathbb{S}^2)}{\pi n}} = \sqrt{\frac{4}{n}}.$$

It follows that $\bar{\mathfrak{s}}(G) \ge \frac{\sqrt{n}}{4}$.

Observe that the three coordinate projections $\mathbb{R}^3 \to \mathbb{R}$ are all Lipschitz with respect to dist_ω, and one of them contributes at least a $1/3$ fraction to the sum (6). We conclude that $\bar{\mathfrak{s}}_{\mathrm{obs}}(G) \ge \frac{\sqrt{n}}{12}$. Combined with Theorem 6, this yields a proof of the Lipton-Tarjan separator theorem [18]. Similar proofs of the separator theorem based on circle packings are known (see [23]), and this one is not new (certainly it was known to the authors of [26]).

We will prove Theorem 1 in two steps: By first giving a lower bound $\bar{\mathfrak{s}}(G) \gtrsim n/\sqrt{m}$ and then establishing $\bar{\mathfrak{s}}_{\mathrm{obs}}(G) \gtrsim \bar{\mathfrak{s}}(G)$.

For the first step, we follow [21, 4, 2]. The optimization (5) is a linear program, and the dual optimization is a maximum multi-flow problem in G. Matoušek shows that a low-congestion multi-flow can be used to draw the complete graph in the plane with few edge crossings. Since this is impossible by a simple double-counting argument, one concludes that there is no low-congestion flow, providing a lower bound on $\bar{\mathfrak{s}}(G)$ via LP duality. We extend this argument to rigs over K_h-minor-free graphs using the flow crossing framework of [2].

1.4 Spread vs. observable spread

Our major departure from [21] comes in the second step: Rounding a fractional separator to an integral separator by establishing that $\bar{\mathfrak{s}}_{\mathrm{obs}}(G) \geq C_h \cdot \bar{\mathfrak{s}}(G)$ when G is a rig over a K_h-minor-free graph. Matoušek used the following result that holds for any metric space. It follows easily from the methods of [3] or [17] (see also [20, Ch. 15]).

▶ **Theorem 9.** *For any finite metric space (X, d) with $|X| \geq 2$, it holds that*

$$\mathfrak{s}_{\mathrm{obs}}(X, d) \geq \frac{\mathfrak{s}(X, d)}{O(\log |X|)} .$$

In particular, for any graph G on $n \geq 2$ vertices,

$$\bar{\mathfrak{s}}_{\mathrm{obs}}(G) \geq \frac{\bar{\mathfrak{s}}(G)}{O(\log n)} .$$

Instead of using the preceding result, we employ the graph partitioning method of Klein, Plotkin, and Rao [11]. Those authors present an iterative process for repeatedly partitioning a metric graph G until the diameter of the remaining components is bounded. If the partitioning process fails, they construct a K_h minor in G.

Since rigs over K_h-minor-free graphs do not necessarily exclude any minors, we need to construct a different sort of forbidden structure. This is the role that Lemma 5 plays in [15]. In order for the argument to work, it is essential that we construct induced partitions: We remove a subset of the vertices which induces a partitioning of the remainder into connected components. After constructing a suitable random partition of G, standard methods from metric embedding theory allow us to conclude

1.5 Eigenvalues and L^2-extremal spread

The methods presented here can be used to control eigenvalues of the discrete Laplacian on rigs. Consider the linear space $\mathbb{R}^{V_G} = \{f : V_G \to \mathbb{R}\}$. Let $\mathcal{L}_G : \mathbb{R}^{V_G} \to \mathbb{R}^{V_G}$ be the symmetric, positive semi-definite linear operator given by

$$\mathcal{L}_G f(v) = \sum_{u:\{u,v\} \in E_G} (f(v) - f(u)) .$$

Let $0 = \lambda_0(G) \leq \lambda_1(G) \leq \cdots \leq \lambda_{|V_G|-1}(G)$ denote the spectrum of $\mathcal{L}_G$.

Define the *L^p-extremal spread* of G as

$$\bar{\mathfrak{s}}_p(G) = \max_{\omega : V_G \to \mathbb{R}_+} \left\{ \frac{1}{|V_G|^2} \sum_{u,v \in V_G} \mathrm{dist}_\omega(u, v) : \|\omega\|_{L^p(V_G)} \leq 1 \right\} . \tag{7}$$

In [2], the L^2-extremal spread is used to give upper bounds on the first non-trivial eigenvalue of graphs that exclude a fixed minor. In [10], a stronger property of conformal metrics is used to bound the higher eigenvalues as well. Roughly speaking, to control the kth eigenvalue, one requires a conformal metric $\omega : V_G \to \mathbb{R}_+$ such that the spread on every subset of size $\geq |V_G|/k$ remains large. Combining their main theorems with our methods yields the following.

▶ **Theorem 10.** *Suppose that $G \in \mathsf{rig}(G_0)$ and G_0 excludes K_h as a minor for some $h \geq 3$. If $d_{\max}$ is the maximum degree of G, then for any $k = 1, 2, \ldots, |V_G| - 1$, it holds that*

$$\lambda_k(G) \leq O(d_{\max}^2 h^6 \log h) \frac{k}{|V_G|} .$$

In particular, the bound on $\lambda_1(G)$ shows that if $d_{\max}(G) \leq O(1)$, then recursive spectral partitioning (see [26]) finds an $O(\sqrt{n})$-vertex balanced separator in G.

1.6 Additional applications

Treewidth approximations. Bounding $\bar{\mathfrak{s}}_{\mathrm{obs}}(G)$ for rigs over K_h-minor-free graphs leads to some additional applications. Combined with the rounding algorithm implicit in Theorem 6 (and explicit in [4]), this yields an $O(h^2)$-approximation algorithms for the vertex uniform Sparsest Cut problem. In particular, it follows that if $G \in \mathsf{rig}(G_0)$ and G_0 excludes K_h as a minor, then there is a polynomial-time algorithm that constructs a tree decomposition of G with treewidth $O(h^2\mathrm{tw}(G))$, where $\mathrm{tw}(G)$ is the treewidth of G. This result appears new even for string graphs. We refer to [4].

Lipschitz extension. Our results on padded decomposability of conformal metrics on string graphs combine with the Lipschitz extension theory of [16] to show the following. Suppose that (G, ω) is a conformal graph, where G is a rig over some K_h-minor free graph. Then for every Banach space Z, subset $S \subseteq V_G$, and L-Lipschitz mapping $f : S \to Z$, there is an $O(h^2 L)$-Lipschitz extension $\tilde{f} : V_G \to Z$ with $\tilde{f}|_S = f$. See [19] for applications to flow and cut sparsifiers in such graphs.

Acknowledgements. The author thanks Noga Alon, Nati Linial, and Laci Lovász for helpful discussions, Janos Pach for emphasizing Jirka's near-optimal bound for separators in string graphs, and the organizers of the "Mathematics of Jiří Matoušek" conference, where much of this work was carried out.

References

1 Noga Alon, Paul Seymour, and Robin Thomas. A separator theorem for nonplanar graphs. *J. Amer. Math. Soc.*, 3(4):801–808, 1990.

2 Punyashloka Biswal, James R. Lee, and Satish Rao. Eigenvalue bounds, spectral partitioning, and metrical deformations via flows. *J. ACM*, 57(3):Art. 13, 23, 2010. Prelim. version in *FOCS 2008.* doi:10.1145/1706591.1706593.

3 J. Bourgain. On Lipschitz embedding of finite metric spaces in Hilbert space. *Israel J. Math.*, 52(1-2):46–52, 1985.

4 Uriel Feige, MohammadTaghi Hajiaghayi, and James R. Lee. Improved approximation algorithms for minimum weight vertex separators. *SIAM J. Comput.*, 38(2):629–657, 2008. doi:10.1137/05064299X.

5 Jacob Fox and János Pach. A separator theorem for string graphs and its applications. *Combin. Probab. Comput.*, 19(3):371–390, 2010. doi:10.1017/S0963548309990459.

6 Jacob Fox and János Pach. Applications of a new separator theorem for string graphs. *Combin. Probab. Comput.*, 23(1):66–74, 2014. doi:10.1017/S0963548313000412.

7 Jacob Fox, János Pach, and Csaba D. Tóth. A bipartite strengthening of the crossing lemma. *J. Combin. Theory Ser. B*, 100(1):23–35, 2010. doi:10.1016/j.jctb.2009.03.005.

8 Misha Gromov. *Metric structures for Riemannian and non-Riemannian spaces.* Modern Birkhäuser Classics. Birkhäuser Boston, Inc., Boston, MA, english edition, 2007. Based on the 1981 French original, With appendices by M. Katz, P. Pansu and S. Semmes, Translated from the French by Sean Michael Bates.

9 Joseph Hersch. Quatre propriétés isopérimétriques de membranes sphériques homogènes. *C. R. Acad. Sci. Paris Sér. A-B*, 270:A1645–A1648, 1970.

10 J. Kelner, J. R. Lee, G. Price, and S.-H. Teng. Metric uniformization and spectral bounds for graphs. *Geom. Funct. Anal.*, 21(5):1117–1143, 2011. Prelim. version in *STOC 2009.*

11 Philip N. Klein, Serge A. Plotkin, and Satish Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 682–690, 1993.

12 A. V. Kostochka. The minimum Hadwiger number for graphs with a given mean degree of vertices. *Metody Diskret. Analiz.*, (38):37–58, 1982.

13 Jan Kratochvíl. String graphs. II. Recognizing string graphs is NP-hard. *J. Combin. Theory Ser. B*, 52(1):67–78, 1991. doi:10.1016/0095-8956(91)90091-W.

14 Jan Kratochvíl and Jiří Matoušek. String graphs requiring exponential representations. *J. Combin. Theory Ser. B*, 53(1):1–4, 1991. doi:10.1016/0095-8956(91)90050-T.

15 James R. Lee. Separators in region intersection graphs. Available at arXiv:math/1608.01612, 2016.

16 James R. Lee and Assaf Naor. Extending Lipschitz functions via random metric partitions. *Invent. Math.*, 160(1):59–95, 2005.

17 Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999.

18 Richard J. Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM J. Appl. Math.*, 36(2):177–189, 1979. doi:10.1137/0136016.

19 Konstantin Makarychev and Yury Makarychev. Metric extension operators, vertex sparsifiers and Lipschitz extendability. *Israel J. Math.*, 212(2):913–959, 2016. doi:10.1007/s11856-016-1315-8.

20 J. Matoušek. *Lectures on discrete geometry*, volume 212 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2002.

21 Jiří Matoušek. Near-optimal separators in string graphs. *Combin. Probab. Comput.*, 23(1):135–139, 2014. doi:10.1017/S0963548313000400.

22 Jiří Matoušek. String graphs and separators. In *Geometry, structure and randomness in combinatorics*, volume 18 of *CRM Series*, pages 61–97. Ed. Norm., Pisa, 2015.

23 Gary L. Miller, Shang-Hua Teng, William Thurston, and Stephen A. Vavasis. Separators for sphere-packings and nearest neighbor graphs. *J. ACM*, 44(1):1–29, 1997. doi:10.1145/256292.256294.

24 Marcus Schaefer, Eric Sedgwick, and Daniel Štefankovič. Recognizing string graphs in NP. *J. Comput. System Sci.*, 67(2):365–380, 2003. Special issue on STOC2002 (Montreal, QC). doi:10.1016/S0022-0000(03)00045-X.

25 Marcus Schaefer and Daniel Štefankovič. Decidability of string graphs. *J. Comput. System Sci.*, 68(2):319–334, 2004. doi:10.1016/j.jcss.2003.07.002.

26 Daniel A. Spielman and Shang-Hua Teng. Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications: Special Issue in honor of Miroslav Fiedler*, 421(2–3):284–305, March 2007.

27 Andrew Thomason. An extremal function for contractions of graphs. *Math. Proc. Cambridge Philos. Soc.*, 95(2):261–265, 1984.

Gradient Descent Only Converges to Minimizers: Non-Isolated Critical Points and Invariant Regions

Ioannis Panageas[1] **and Georgios Piliouras**[2]

1 **MIT, USA & Singapore University of Technology and Design, Singapore**
panageasj@gmail.com
2 **Singapore University of Technology and Design, Singapore**
georgios.piliouras@gmail.com

Abstract

Given a twice continuously differentiable cost function f, we prove that the set of initial conditions so that gradient descent converges to saddle points where $\nabla^2 f$ has at least one strictly negative eigenvalue, has (Lebesgue) measure zero, even for cost functions f with non-isolated critical points, answering an open question in [12]. Moreover, this result extends to forward-invariant convex subspaces, allowing for weak (non-globally Lipschitz) smoothness assumptions. Finally, we produce an upper bound on the allowable step-size.

1998 ACM Subject Classification G.1.6 [Numerical Analysis]: Optimization–Gradient methods

Keywords and phrases Gradient Descent, Center-stable manifold, Saddle points, Hessian

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.2

1 Introduction

The interplay between the structure of saddle points and the performance of gradient descent dynamics is a critical and not well understood aspect of non-convex optimization. Despite our incomplete theoretical understanding, in practice, the intuitive nature of the gradient descent method (and more generally gradient-like algorithms[1]) make it a basic tool for attacking non-convex optimization problems for which we have very little understanding of the geometry of their saddle points. In fact, these techniques become particularly useful as the equilibrium structure becomes increasingly complicated, e.g., such as in the cases of nonnegative matrix factorization [11] or congestion/potential games [25], where symmetries in the nature of non-convex optimization problems give rise to continuums of saddle points with complex geometry. In these cases, especially, the simplistic, greedy attitude of the gradient descent method, which is by design agnostic towards the global geometry of the cost function minimized, comes rather handy. As we move forward in time, the cost keeps decreasing and convergence is guaranteed.

This simplicity, however, comes at least seemingly at a significant cost. For example, it is well known that there exist instances where bad initialization of gradient descent converges to saddle points [18]. Despite the existence of such worst case instances in theory, practitioners have been rather successful at applying these techniques across a wide variety of problems [23]. Recently, Lee et. al. [12] have given a rather insightful interpretation of the effectiveness of gradient descent methods in terms of circumventing the saddle equilibrium problem using

[1] A gradient-like system is a system where for each non-equilibrium initial condition the dynamic will move towards a new state whose cost is strictly less than that of the initial state.


© Ioannis Panageas and Georgios Piliouras;
licensed under Creative Commons License CC-BY

tools from topology of dynamical systems. At a glance, the paper argues the following intuitively clear message: *The instability of (locally unstable) saddle points translates to a global phenomenon and the probability of converging to such a saddle point given a randomly chosen (random not over a local neighborhood but over the whole state space) initial condition is zero.*

This message is clear, concise, and satisfying in the sense that it transcribes the practical success of the gradient descent method to a concrete theoretical guarantee. As is usually the case, such high level statements come with an asterisk of necessary technical conditions on the cost function f minimized.

Formally, a cost function f is said to satisfy the "strict saddle" property if each critical point[2] x of f is either a local minimizer, or a strict saddle, i.e., $\nabla^2 f(x)$ has at least one strictly negative eigenvalue. In this case, Lee et. al. argue that if $f : \mathbb{R}^d \to \mathbb{R}$ is a twice continuously differentiable function then gradient descent with constant step-size α (defined by $x_{k+1} = x_k - \alpha \nabla f(x_k)$) with a random initialization and sufficiently small constant step-size converges to a local minimizer or negative infinity almost surely.

Critically, for this result to apply, f is required to have isolated saddle points, ∇f is assumed to be globally L-Lipschitz[3] and the step-size α is taken to be less than $1/L$. These regularity conditions soften somewhat the impact of the statement both theoretically as well as in practice. First, although the assumption of isolated fixed points is indeed generic for abstract classes of cost functions, in several special cases of practical interest where the cost function has some degree of symmetry (e.g., due to scaling invariance) this assumption is not satisfied. For this reason, the question of whether the assumption of isolated equilibria is indeed necessary was explicitly raised in [12]. Moreover, the assumption of global Lipschitz continuity for ∇f is not satisfied even by low degree polynomials (e.g., cubic). Finally, a natural question is how tight is the assumption on the step-size?

In this work we provide answers to all the above questions. We show that the assumption of isolated saddle points is indeed not necessary to argue generic convergence to local minima. To argue this, we need to combine tools from dynamical systems, topology, analysis and optimization theory. Moreover, we show that the globally Lipschitz assumption can be circumvented as long as the domain is convex and forward invariant with respect to gradient descent. This proposition makes our results readily applicable to many standard settings. This extension holds even for non-coercive functions.[4] Finally, using linear algebra and eigenvalue analysis we provide an upper bound on the allowable step-size (for these results to hold).

It is important to clearly state that the focus of this work is more theoretical than applied. Specifically, although GD is guaranteed to converge to local minima this convergence can in the worst case (worst case initial conditions, instances) take exponential time as finding local minima of non-convex functions is an NP-hard problem. Nevertheless, we believe that this combination of topological techniques (standard in continuous-time dynamics) and discrete optimization techniques, besides solving an interesting open question, will hopefully be of use for other algorithmic problems as well. Deep learning, due to the singularity of the Hessian of the loss function in practice [24], is a domain where such techniques could lead to new insights and practical algorithms.

---

[2] x is a critical point of f if $\nabla f(x) = 0$.

[3] That is, f satisfies $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq L \|\mathbf{x} - \mathbf{y}\|_2$.

[4] As pointed out by a referee this extension of Lee et al. is relatively easy when a function is coercive, since the sublevel sets are compact.

1.1 Related work

First-order descent methods can indeed escape strict saddle points when assisted by near isotropic noise. [21] establishes convergence of the Robbins-Monro stochastic approximation to local minimizers for strict saddle functions, whereas [10] establishes convergence to local minima for perturbed versions of multiplicative weights algorithm in generic potential games. Recently, [7] quantified the convergence rate of perturbed stochastic gradient descent to local minima. The addition of isotropic noise can significantly slow down the convergence rate. In contrast, our setting is deterministic and corresponds to the simplest possible discrete-time implementation of gradient descent.

Numerous curvature-based optimization techniques have been developed in order to circumvent saddle points (e.g., trust-region methods [5, 28], modified Newton's method with curvilinear line search [17], cubic regularized Newton's method [19], and saddle-free Newton methods [6]). Unlike gradient descent, these methods have superlinear per-iteration implementation costs, making them impractical for high dimensional settings.

Gradient descent with carefully chosen initial conditions can bypass the problem of local minima altogether and converge to the global minimum for many practical non-convex optimization settings (e.g., dictionary learning [1], latent-variable models [29], matrix completion [9], and phase retrieval [2]). In contrast, we focus on the performance of gradient descent under generic initial conditions. Finally, some recent work has been focusing on the connections between stability and efficiency of fixed points in non-convex optimization (e.g., Gaussian random fields [4]).

Gradient-like dynamics, where the dynamic moves towards states of decreased cost but without necessarily moving in the direction of steepest decrease, is a generalization of gradient dynamics that arise in a number of applications including game theory and mathematical biology. Similar arguments about convergence to local minima for almost all initial conditions have been argued [10, 20, 13] for (variants of) replicator dynamics and multiplicative weights update algorithms when applied to games where the incentives of all agents are closely aligned.[5] From the perspective of biology and specifically evolution, (variants of) replicator/MWUA [3, 16] capture standard models of the evolution of the frequencies of different genotypes within a species (preferential survival of the fittest). By analyzing the properties of local minimum energy states we can derive completely different conclusions about the long term system behavior (in terms e.g., of the resulting genetic diversity) from the ones that follow from analyzing all saddle points [13]. In fact, understanding the properties of local minima raises interesting computational complexity questions [15]. Finally, examining the stability properties of equilibria can help us capture quantitatively the long term behavior of biologically inspired gradient-like systems even under time-evolving fitness landscapes [14]. Given the emergent overlapping interests between these areas and (non-convex) optimization theory, it seems that novel opportunities for cross-fertilization between these research communities arise.

1.2 Organization

In Section 2, we introduce the notation and definitions used throughout the paper and state formally our main theorems. In Section 3, we prove our results establishing the negligible probability of converging to saddle points, addressing the possibility of continuums

[5] Such games are known as potential/congestion games [10] and correspond to games where all agents act as if they share a common cost/potential function that they are trying to minimize.

of equilibria, forward-invariant subspaces, and establishing an upper bound on the step-size. In Section 4, we produce several examples showcasing the effectiveness of our methods. Finally, we conclude in Section 5 by suggesting directions for future work.

2 Preliminaries

Notation: We use boldface letters, e.g., $\mathbf{x}$, to denote column vectors. We denote by $\mathrm{sp}(A), \|A\|_2$ the spectral radius and spectral norm of a symmetric matrix A respectively. We also use $\|\mathbf{x}\|_2$ for the ℓ_2 norm of vector $\mathbf{x}$. By $\nabla^2 f(\mathbf{x})$ we denote the Hessian of a twice differentiable function $f : \mathcal{E} \to \mathbb{R}$, for some set $\mathcal{E} \subseteq \mathbb{R}^N$.

Assume a minimization problem of the form $\min_{\mathbf{x}\in\mathbb{R}^N} f(\mathbf{x})$ where $f : \mathbb{R}^N \to \mathbb{R}$ is a twice continuously differentiable function. Gradient descent is one of the most well-known algorithms (discrete dynamical system) to attack this generic optimization problem. It is defined by the equations below:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha\nabla f(\mathbf{x}_k),$$

or equivalently $\mathbf{x}_{k+1} = g(\mathbf{x}_k)$ with $g(\mathbf{x}) = \mathbf{x} - \alpha\nabla f(\mathbf{x})$, $g : \mathbb{R}^N \to \mathbb{R}^N$ and $\alpha > 0$.

It is easy to see that the fixed points of the dynamical system $\mathbf{x}_{k+1} = g(\mathbf{x}_k)$ are exactly the points $\mathbf{x}$ so that $\nabla f(\mathbf{x}) = \mathbf{0}$, called *critical points or equilibria*. The set of local minima of f is a subset of the set of critical points of f. These two sets do not coincide and this poses a serious obstacle for proving strong theoretical guarantees for gradient descent, since the dynamics may converge to a critical point which is not a local minimum, called a *saddle point*.

Lee et al. [12] argue, under technical conditions which include the assumption of isolated critical points, that the set of initial conditions that converge to *strict saddle points* is a zero measure set (for definition of strict saddle, see Definition 1). The paper leaves as an open question whether the condition of isolated equilibria is necessary. We prove that the set of initial conditions that converge to a strict saddle point is a zero measure set even in the case of non-isolated critical points[6]. Furthermore, one of the conditions for f is that ∇f is globally Lipschitz, which implies that the second derivative of f is bounded, i.e., there exists a $\beta > 0$ such that for all $\mathbf{x}$ we have $\left\|\nabla^2 f(\mathbf{x})\right\|_2 \leq \beta$. However, even third degree polynomial functions are not globally Lipschitz. We provide a theorem which can circumvent this assumption as long as the domain $\mathcal{S}$ is *forward or positively invariant* with respect to g, i.e., $g(\mathcal{S}) \subseteq \mathcal{S}$. Finally, we provide an easy upper bound on the step-size α, via eigenvalue analysis of the Jacobian of g, i.e., $I - \alpha\nabla^2 f(\mathbf{x})$.

Below we give some necessary definitions as appeared in Lee et al. [12].

▶ **Definition 1.**

- A point $\mathbf{x}^*$ is a critical point of f if $\nabla f(\mathbf{x}^*) = \mathbf{0}$. We denote by $C = \{\mathbf{x} : \nabla f(\mathbf{x}) = \mathbf{0}\}$ the set of critical points (can be uncountably many).
- A critical point $\mathbf{x}^*$ is isolated if there is a neighborhood U around $\mathbf{x}^*$ and $\mathbf{x}^*$ is the only critical point in U.[7] Otherwise is called non-isolated.
- A critical point $\mathbf{x}^*$ of f is a saddle point if for all neighborhoods U around $\mathbf{x}^*$ there are $\mathbf{y}, \mathbf{z} \in U$ such that $f(\mathbf{z}) \leq f(\mathbf{x}^*) \leq f(\mathbf{y})$.

[6] Our arguments hence allow for cost functions f's with uncountably many critical points.
[7] If the critical points are isolated then they are countably many or finite.

- A critical point $\mathbf{x}^*$ of f is a strict saddle if $\lambda_{\min}(\nabla^2 f(\mathbf{x}^*)) < 0$ (minimum eigenvalue of matrix $\nabla^2 f(\mathbf{x}^*)$ is negative).
- A set $\mathcal{S}$ is called forward or positively invariant with respect to some function $h : \mathcal{E} \to \mathbb{R}^N$ with $\mathcal{S} \subseteq \mathcal{E} \subseteq \mathbb{R}^N$ if $h(\mathcal{S}) \subseteq \mathcal{S}$.

2.1 Main Results

In [12], the steps of the proof of their result are the following: Under the regularity assumption that ∇f is globally Lipschitz with some Lipschitz constant L, Lee et al. are able to show that $g(\mathbf{x}) = \mathbf{x} - \alpha \nabla f(\mathbf{x})$ is a diffeomorphism for $\alpha < 1/L$. Afterwards, using the center-stable manifold theorem (see theorem 8), they show that the set of initial conditions so that g converges to saddle points has measure zero under the assumption that the critical points are isolated. We generalize their result for non-isolated critical points, answering one of their open questions (see also the example in Section 4.1, where there is a line of critical points).

▶ **Theorem 2.** [**Non-isolated**] *Let $f : \mathbb{R}^N \to \mathbb{R}$ be a twice continuously differentiable function and $\sup_{\mathbf{x}\in\mathbb{R}^N} \left\|\nabla^2 f(\mathbf{x})\right\|_2 \leq L < \infty$. The set of initial conditions $\mathbf{x} \in \mathbb{R}^N$ so that gradient descent with step-size $0 < \alpha < 1/L$ converges to a strict saddle point is of (Lebesgue) measure zero, without the assumption that critical points are isolated.*

We can prove a stronger version of the theorem above, circumventing the globally Lipschitz condition for domains which are forward invariant (see also the example in Section 4.2).

▶ **Theorem 3.** [**Non-isolated, forward invariant**] *Let $f : \mathcal{S} \to \mathbb{R}$ be twice continuously differentiable in an open convex set $\mathcal{S} \subseteq \mathbb{R}^N$ and $\sup_{\mathbf{x}\in\mathcal{S}} \left\|\nabla^2 f(\mathbf{x})\right\|_2 \leq L < \infty$. If $g(\mathcal{S}) \subseteq \mathcal{S}$ (where $g(\mathbf{x}) = \mathbf{x} - \alpha\nabla f(\mathbf{x})$) then the set of initial conditions $\mathbf{x} \in \mathcal{S}$ so that gradient descent with step-size $0 < \alpha < 1/L$ converges to a strict saddle point is of (Lebesgue) measure zero, without the assumption that critical points are isolated.*

Finally, via eigenvalue analysis of $I - \alpha\nabla^2 f(\mathbf{x})$, we can find upper bounds on the step-size of gradient descent. A straightforward theorem is the following:

▶ **Theorem 4.** [**Upper bound on step-size**] *Let f be twice continuously differentiable in an open set $\mathcal{S} \subseteq \mathbb{R}^N$ and $\mathcal{C}^*$ be the set of local minima. Assume also that $\gamma < \inf_{\mathbf{x}\in\mathcal{C}^*} \left\|\nabla^2 f(\mathbf{x})\right\|_2 < \infty$. A necessary condition so that gradient descent converges to local minima for all but (Lebesgue) measure zero initial conditions in $\mathcal{S}$ is that the step-size satisfies $\alpha < \frac{2}{\gamma}$.*

3 Proving the theorems

Before we proceed with the proofs, let us argue that Theorem 3 is a generalization of Theorem 2. This can be checked by setting $\mathcal{S} := \mathbb{R}^N$ and observing that $g(\mathbb{R}^N) \subseteq \mathbb{R}^N$. We continue with the proofs of Theorems 3 and 4.

3.1 Proof of Theorem 3

In this section, we prove Theorem 3. We start by showing (for completeness) that the assumptions of Theorem 3 imply that $\nabla f(\mathbf{x})$ is Lipschitz in $\mathcal{S}$.

▶ **Lemma 5.** *Let $f : \mathcal{S} \to \mathbb{R}$ where $\mathcal{S}$ is an open convex set and f be twice continuously differentiable in $\mathcal{S}$. Also assume that $\sup_{\mathbf{x}\in\mathcal{S}} \left\|\nabla^2 f(\mathbf{x})\right\|_2 \leq L < \infty$. Then ∇f satisfies the Lipschitz condition in $\mathcal{S}$ with Lipschitz constant L.*

Proof. Let $\mathbf{x}, \mathbf{y} \in \mathcal{S}$ (column vectors) and define the function $H : [0,1] \to \mathbb{R}^N$ as $H(t) = \nabla f(\mathbf{x} + t(\mathbf{y} - \mathbf{x}))$. By the chain rule we get that $H'(t) := \frac{dH}{dt} = (\nabla^2 f(\mathbf{x} + t(\mathbf{y} - \mathbf{x}))) \cdot (\mathbf{y} - \mathbf{x})$. It holds that

$$\begin{aligned}
\|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x})\|_2 = \left\| \int_0^1 H'(t) dt \right\|_2 &\leq \int_0^1 \|H'(t)\|_2 \, dt \\
&= \int_0^1 \left\| (\nabla^2 f(\mathbf{x} + t(\mathbf{y} - \mathbf{x})))(\mathbf{y} - \mathbf{x}) \right\|_2 dt \\
&\leq \int_0^1 \left\| \nabla^2 f(\mathbf{x} + t(\mathbf{y} - \mathbf{x})) \right\|_2 \|\mathbf{y} - \mathbf{x}\|_2 \, dt \\
&\leq \int_0^1 L \|\mathbf{y} - \mathbf{x}\|_2 \, dt = L \|\mathbf{y} - \mathbf{x}\|_2 .
\end{aligned}$$

◀

▶ Remark. From Schwarz's theorem we get that $\nabla^2 f(\mathbf{x})$ is symmetric for $\mathbf{x} \in \mathcal{S}$, hence $\left\|\nabla^2 f(\mathbf{x})\right\|_2 = \text{sp}(\nabla^2 f(\mathbf{x}))$.

The assumption that $\sup_{\mathbf{x} \in \mathcal{S}} \left\|\nabla^2 f(\mathbf{x})\right\|_2 \leq L < \infty$ implies that $\nabla f(x)$ is Lipschitz with constant L in the convex set $\mathcal{S}$, as stated by Lemma 5. We show that the converse holds as well, i.e., the Lipschitz condition for $\nabla f(\mathbf{x})$ with constant L in the main theorem in Lee et al. implies $\left\|\nabla^2 f(\mathbf{x})\right\|_2 \leq L$ for all $\mathbf{x} \in \mathcal{S}$ and hence the assumption in our Theorems 2, 3 that $\sup_{\mathbf{x} \in \mathcal{S}} \left\|\nabla^2 f(\mathbf{x})\right\|_2 \leq L$ is satisfied.

▶ **Lemma 6.** *Let $f : \mathcal{S} \to \mathbb{R}$ where $\mathcal{S} \subseteq \mathbb{R}^N$ is an open convex set and f is twice continuously differentiable in $\mathcal{S}$. Assume $\nabla f(\mathbf{x})$ is Lipschitz with constant L in $\mathcal{S}$ then it holds $\sup_{\mathbf{x} \in \mathcal{S}} \left\|\nabla^2 f(\mathbf{x})\right\|_2 \leq L$.*

Proof. Fix an $\epsilon > 0$. By Taylor's theorem since f is twice differentiable with respect to some point $\mathbf{x}$ it holds that

$$\begin{aligned}
\|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x})\|_2 &\geq \left\| (\nabla^2 f(\mathbf{x}))(\mathbf{y} - \mathbf{x}) \right\|_2 - o(\|\mathbf{y} - \mathbf{x}\|_2) \\
&\geq \left\| (\nabla^2 f(\mathbf{x}))(\mathbf{y} - \mathbf{x}) \right\|_2 - \epsilon \|\mathbf{y} - \mathbf{x}\|_2
\end{aligned}$$

for $\mathbf{y}$ sufficiently close to $\mathbf{x}$ (depends on ϵ). Therefore under the Lipschitz assumption we get that there exists a closed neighborhood $U(\epsilon)$ of $\mathbf{x}$ so that for all $\mathbf{y} \in U$ we get

$$\left\| (\nabla^2 f(\mathbf{x}))(\mathbf{x} - \mathbf{y}) \right\|_2 \leq \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 + \epsilon \|\mathbf{y} - \mathbf{x}\|_2 \leq (L + \epsilon) \|\mathbf{x} - \mathbf{y}\|_2 . \tag{1}$$

We consider a closed ball B subset of U, with center $\mathbf{x}$ and radius r (in ℓ_2) and set $\mathbf{z} = \mathbf{x} - \mathbf{y}$. It is true that $\left\|\nabla^2 f(\mathbf{x})\right\|_2 = \sup_{\|\mathbf{z}\|_2 = r} \frac{\left\|(\nabla^2 f(\mathbf{x}))\mathbf{z}\right\|_2}{\|\mathbf{z}\|_2}$ by definition of spectral norm, scaled so that the length of the vectors is exactly r. Using 1 we get that $\left\|\nabla^2 f(\mathbf{x})\right\|_2 \leq L + \epsilon$. Since ϵ is arbitrary, we get that $\left\|\nabla^2 f(\mathbf{x})\right\|_2 \leq L$. We conclude that $\sup_{\mathbf{x} \in \mathcal{S}} \left\|\nabla^2 f(\mathbf{x})\right\|_2 \leq L$. ◀

Lemmas 5 and 6 (which are provided for completeness) show that the smoothness assumptions in Lee et al. paper are equivalent to ours. We use the condition on the spectral norm of the matrix $\nabla^2 f(\mathbf{x})$ so that we can work with the eigenvalues in our theorems (e.g., in Remark 3.1 the spectral norm coincides with spectral radius for $\nabla^2 f(\mathbf{x})$). Below we prove that the update rule of gradient descent, i.e., function g is a diffeomorphism under the assumptions of Theorem 3 (similar approach appeared in [12]).

▶ **Lemma 7.** *Under the assumptions of Theorem 3, function g is a diffeomorphism in $\mathcal{S}$.*

Proof. First we prove that g is a injective. We follow the same argument as in [12]. Suppose $g(\mathbf{y}) = g(\mathbf{x})$, thus $\mathbf{y} - \mathbf{x} = \alpha(\nabla f(\mathbf{y}) - \nabla f(\mathbf{x}))$. We assume that $\mathbf{x} \neq \mathbf{y}$ and we will reach contradiction. From Lemma 5 we get $\|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x})\|_2 \leq L \|\mathbf{y} - \mathbf{x}\|_2$ and hence $\|\mathbf{x} - \mathbf{y}\|_2 \leq \alpha L \|\mathbf{y} - \mathbf{x}\|_2 < \|\mathbf{y} - \mathbf{x}\|_2$ since $\alpha L < 1$ (contradiction).

We continue by showing that g is a local diffeomorphism. Observe that the Jacobian of g is $I - \alpha\nabla^2 f(\mathbf{x})$. It suffices to show that $\alpha\nabla^2 f(\mathbf{x})$ has no eigenvalue which is 1, because this implies matrix $I - \alpha\nabla^2 f(\mathbf{x})$ is invertible. As long as $I - \alpha\nabla^2 f(\mathbf{x})$ is invertible, from Inverse Function Theorem (see [27]) follows that g is a local diffeomorphism. Finally, since g is injective, the inverse g^{-1} is well defined and since g is a local diffeomorphism in $\mathcal{S}$, it follows that g^{-1} is smooth in $\mathcal{S}$. Therefore g is a diffeomorphism.

Let λ be an eigenvalue of $\nabla^2 f(\mathbf{x})$. Then $|\lambda| \leq \text{sp}(\nabla^2 f(\mathbf{x})) = \left\|\nabla^2 f(\mathbf{x})\right\|_2 \leq L$ where the equality comes from Remark 3.1 and first and last inequalities are satisfied by assumption. Therefore $\alpha\nabla^2 f(\mathbf{x})$ has as eigenvalue $\alpha\lambda$ and $|\alpha\lambda| \leq \alpha L < 1$. Thus all eigenvalues of $\alpha\nabla^2 f(\mathbf{x})$ are less than 1 in absolute value and the proof is complete. ◀

Proof of Theorem 3. We will use the Center-stable manifold theorem since $g(\mathbf{x}) = \mathbf{x} - \alpha\nabla f(\mathbf{x})$ is a diffeomorphism, where $\sup_{\mathbf{x}\in\mathcal{S}} \left\|\nabla^2 f(\mathbf{x})\right\|_2 \leq L$ and $\alpha < 1/L$. A modification of the proof of Theorem 3 appeared in [20] and [13] for replicator dynamics (not gradient descent).

▶ **Theorem 8** (Center and Stable Manifolds, p. 65 of [26]). *Let $\mathbf{p}$ be a fixed point for the C^r local diffeomorphism $h : U \to \mathbb{R}^n$ where $U \subset \mathbb{R}^n$ is an open neighborhood of $\mathbf{p}$ in $\mathbb{R}^n$ and $r \geq 1$. Let $E^s \oplus E^c \oplus E^u$ be the invariant splitting of $\mathbb{R}^n$ into generalized eigenspaces of $Dh(\mathbf{p})$[8] corresponding to eigenvalues of absolute value less than one, equal to one, and greater than one. To the $Dh(\mathbf{p})$ invariant subspace $E^s \oplus E^c$ there is an associated local h invariant C^r embedded disc W^{sc}_{loc} of dimension $dim(E^s \oplus E^c)$, and ball B around $\mathbf{p}$ such that:*

$$h(W^{sc}_{loc}) \cap B \subset W^{sc}_{loc}. \text{ If } h^n(\mathbf{x}) \in B \text{ for all } n \geq 0, \text{ then } \mathbf{x} \in W^{sc}_{loc}. \tag{2}$$

From this point on our approach deviates significantly from that of [12] and new ideas and tools need to be introduced.

Let $\mathbf{r}$ be a critical point of function $f(\mathbf{x})$ and $B_{\mathbf{r}}$ be the (open) ball that is derived from Theorem 8. We consider the union of these balls

$$A = \cup_{\mathbf{r}} B_{\mathbf{r}}.$$

The following property for $\mathbb{R}^N$ holds:

▶ **Theorem 9** (Lindelőf's lemma [8]). *For every open cover there is a countable subcover.*

Therefore due to Lindelőf's lemma, we can find a countable subcover for A, i.e., there exist fixed points $\mathbf{r}_1, \mathbf{r}_2, \ldots$ such that $A = \cup_{m=1}^{\infty} B_{\mathbf{r}_m}$. If gradient descent converges to a strict saddle point, starting from a point $\mathbf{v} \in \mathcal{S}$, there must exist a t_0 and m so that $g^t(\mathbf{v}) \in B_{\mathbf{r}_m}$ for all $t \geq t_0$. From Theorem 8 we get that $g^t(\mathbf{v}) \in W^{sc}_{loc}(\mathbf{r}_m) \cap \mathcal{S}$ where we used the fact that $g(\mathcal{S}) \subseteq \mathcal{S}$ (from assumption forward invariant), namely the trajectory remains in $\mathcal{S}$ for all times [9]. By setting $D_1(\mathbf{r}_m) = g^{-1}(W^{sc}_{loc}(\mathbf{r}_m) \cap \mathcal{S})$ and $D_{i+1}(\mathbf{r}_m) = g^{-1}(D_i(\mathbf{r}_m) \cap \mathcal{S})$ we get

[8] Jacobian of h evaluated at $\mathbf{p}$.
[9] $W^{sc}_{loc}(\mathbf{r}_m)$ denotes the center stable manifold of fixed point $\mathbf{r}_m$

that $\mathbf{v} \in D_t(\mathbf{r}_m)$ for all $t \geq t_0$. Hence the set of initial points in $\mathcal{S}$ so that gradient descent converges to a strict saddle point is a subset of

$$P = \cup_{m=1}^{\infty} \cup_{t=0}^{\infty} D_t(\mathbf{r}_m)). \tag{3}$$

Since $\mathbf{r}_m$ is strict saddle point, the Jacobian $I - \alpha\nabla^2 f(\mathbf{x})$ has an eigenvalue greater than 1, namely the dimension of the unstable eigenspace satisfies $dim(E^u) \geq 1$, and therefore the dimension of $W^{sc}_{loc}(\mathbf{r}_m)$ is at most $N-1$. Thus, the set $W^{sc}_{loc}(\mathbf{r}_m) \cap \mathcal{S}$ has Lebesgue measure zero in $\mathbb{R}^N$. Finally since g is a diffeomorphism (from Lemma 7), g^{-1} is continuously differentiable and thus it is locally Lipschitz (see [22] p.71). Therefore using Lemma 10 which we state below, g^{-1} preserves the null-sets and hence (by induction) $D_i(\mathbf{r}_m)$ has measure zero for all i. Thereby we get that P is a countable union of measure zero sets, i.e., is measure zero as well and the claim of Theorem 3 follows. ◀

▶ **Lemma 10.** *Let $h : \mathcal{S} \to \mathbb{R}^m$ be a locally Lipschitz function with $\mathcal{S} \subseteq \mathbb{R}^m$ then h is null-set preserving, i.e., for $E \subset \mathcal{S}$ if E has measure zero then $h(E)$ has also measure zero.*

Proof. The lemma is quite standard, but we provide a proof for completeness. Let B_γ be an open ball such that $\|h(\mathbf{y}) - h(\mathbf{x})\| \leq K_\gamma \|\mathbf{y} - \mathbf{x}\|$ for all $\mathbf{x}, \mathbf{y} \in B_\gamma$. We consider the union $\cup_\gamma B_\gamma$ which cover $\mathbb{R}^m$ by the assumption that h is locally Lipschitz. By Lindelőf's lemma we have a countable subcover, i.e., $\cup_{i=1}^{\infty} B_i$. Let $E_i = E \cap B_i$. We will prove that $h(E_i)$ has measure zero. Fix an $\epsilon > 0$. Since $E_i \subset E$, we have that E_i has measure zero, hence we can find a countable cover of open balls $C_1, C_2, ...$ for E_i, namely $E_i \subset \cup_{j=1}^{\infty} C_j$ so that $C_j \subset B_i$ for all j and also $\sum_{j=1}^{\infty} \mu(C_j) < \frac{\epsilon}{K_i^m}$. Since $E_i \subset \cup_{j=1}^{\infty} C_j$ we get that $h(E_i) \subset \cup_{j=1}^{\infty} h(C_j)$, namely $h(C_1), h(C_2), ...$ cover $h(E_i)$ and also $h(C_j) \subset h(B_i)$ for all j. Assuming that ball $C_j \equiv B(\mathbf{x}, r)$ (center $\mathbf{x}$ and radius r) then it is clear that $h(C_j) \subset B(h(\mathbf{x}), K_i r)$ (h maps the center $\mathbf{x}$ to $h(\mathbf{x})$ and the radius r to $K_i r$ because of Lipschitz assumption). But $\mu(B(h(\mathbf{x}), K_i r)) = K_i^m \mu(B(\mathbf{x}, r)) = K_i^m \mu(C_j)$, therefore $\mu(h(C_j)) \leq K_i^m \mu(C_j)$ and so we conclude that

$$\mu(h(E_i)) \leq \sum_{j=1}^{\infty} \mu(h(C_j)) \leq K_i^m \sum_{j=1}^{\infty} \mu(C_j) < \epsilon$$

Since ϵ was arbitrary, it follows that $\mu(h(E_i)) = 0$. To finish the proof, observe that $h(E) = \cup_{i=1}^{\infty} h(E_i)$ therefore $\mu(h(E)) \leq \sum_{i=1}^{\infty} \mu(h(E_i)) = 0$. ◀

A straightforward application of Theorem 3 is the following:

▶ **Corollary 11.** *Assume that the conditions of Theorem 3 are satisfied and all saddle points of f are strict. Additionally, let ν be a prior measure with support $\mathcal{S}$ which is absolutely continuous with respect to Lebesgue measure, and assume $\lim_{k\to\infty} g^k(x)$ exists[10] for all $\mathbf{x}$ in $\mathcal{S}$. Then*

$$\mathbb{P}_\nu[\lim_k g^k(\mathbf{x}) = \mathbf{x}^*] = 1,$$

where $\mathbf{x}^*$ is a local minimum.

Proof. Since the set of initial conditions whose limit point is a (strict) saddle point is a measure zero set and we have assumed $\lim_{k\to\infty} g^k(x)$ exists for all initial conditions in $\mathcal{S}$ then the probability of converging to a local minimizer is 1. ◀

[10] g^k denotes the composition of g with itself k times.

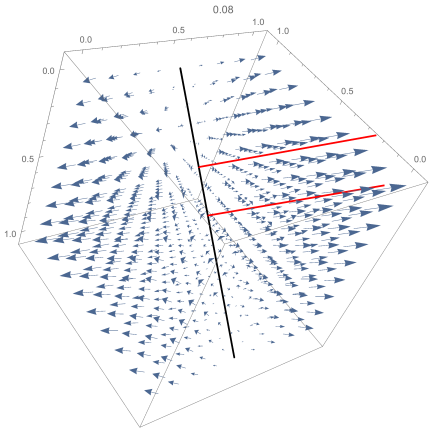


Figure 1 Example that satisfies the assumptions of Theorem 2. The black line represent critical points of f, all of which are strict. The red lines correspond to diverging trajectories of gradient descent with small step size.

▶ Remark. Arguing that $\lim_k g^k(\mathbf{x})$ exists follows from standard arguments in several settings of interest (e.g for analytic functions f that satisfy (Lojasiewicz Gradient Inequality)), see paper [12] and references therein.

The importance of Theorem 3 will become clear in the examples of Section 4. Specifically, in the example of Section 4.2, the function is not globally Lipschitz (we use the example that appears in [12]), nevertheless Theorem 3 applies and thus we have convergence to local minimizers with probability 1. In the example of Section 4.1 we see that simple functions may have non-isolated critical points.

3.2 Proof of Theorem 4

Proof. We proceed by contradiction. Consider any local minimum $\mathbf{x}^*$, and by assumption we get that $\text{sp}(\nabla^2 f(\mathbf{x}^*)) > \gamma$. Let $\alpha \geq \frac{2}{\gamma}$. Therefore the Jacobian $I - \alpha\nabla^2 f(\mathbf{x}^*)$ of g at $\mathbf{x}^*$ has spectral radius greater than 1 since $\text{sp}(I - \alpha\nabla^2 f(\mathbf{x}^*)) \geq \text{sp}(\alpha\nabla^2 f(\mathbf{x}^*)) - 1 > \alpha\gamma - 1 \geq 1$. This implies that the fixed point $\mathbf{x}^*$ of g is (Lyapunov) unstable. Since this is true for every local minimum, it cannot be true that gradient descent converges with probability 1 to local minima. ◀

4 Examples

4.1 Example for non-isolated critical points

Consider the simple example of the cost function $f : \mathbb{R}^3 \to \mathbb{R}$ with $f(x, y, z) = 2xy + 2xz - 2x - y - z$. Its gradient is $\nabla(f) = (2y + 2z - 2, 2x - 1, 2x - 1)$. Naturally, its saddle points correspond exactly to the line $(1/2, w, 1 - w)$ for $w \in \mathbb{R}$ and by computing their (common) eigenvalues we establish that they are all strict saddles (their minimum eigenvalue is $-2\sqrt{2}$). As we expect from our analysis effectively no trajectories converge to them (instead the value of practically all trajectories goes to $-\infty$). We plot in red some sample trajectories for small enough step sizes, starting in the local neighborhood of the equilibrium set.

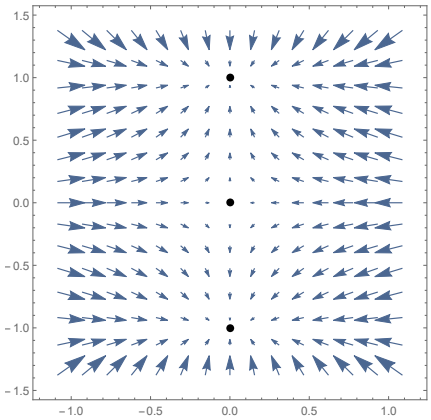


Figure 2 Example that satisfies the assumptions of Theorem 3. The three black dots represent the critical points. Function f is not Lipschitz.

4.2 Example for forward invariant set

We use the same function as in Lee et al. $f(x,y) = \frac{x^2}{2} + \frac{y^4}{4} - \frac{y^2}{2}$. As argued in previous sections, f is not globally Lipschitz so the main result in [12] cannot be applied here. We will use our Theorem 3 which talks about forward invariant domains.

The critical points of f are $(0,0),(0,1),(0,-1)$. $(0,0)$ is a strict saddle point and the other two are local minima. Observe that the Hessian $\nabla^2 f(x,y)$ is

$$J = \begin{pmatrix} 1 & 0 \\ 0 & 3y^2 - 1 \end{pmatrix}.$$

For $\mathcal{S} = (-1,1) \times (-2,2)$, so we get that $\sup_{(x,y)\in\mathcal{S}} \left\|\nabla^2 f(x,y)\right\|_2 \leq 11$ (for $y = 2$ gets the maximum value). We choose $\alpha = \frac{1}{12} < \frac{1}{11}$, and we have $g(x,y) = ((1-\alpha)x, (1+\alpha)y - \alpha y^3) = (\frac{11x}{12}, \frac{13y}{12} - \frac{y^3}{12})$. It is not difficult to see that $g(\mathcal{S}) \subseteq \mathcal{S}$ (easy calculations). The assumptions of Theorem 3 are satisfied, hence it is true that the set of initial conditions in $\mathcal{S}$ so that gradient descent converges to $(0,0)$ has measure zero. Moreover, by Corollary 11 it holds that if the initial condition is taken (say) uniformly at random in $\mathcal{S}$, then gradient descent converges to $(0,1),(0,-1)$ with probability 1. The figure below makes the claim clear, i.e. the set of initial conditions so that gradient descent converges to $(0,0)$ lie on the axis $y = 0$, which is of measure zero in $\mathbb{R}^2$. For all other starting points, gradient descent converges to local minima. Finally, from the figure one can see that $\mathcal{S}$ is forward invariant.

4.3 Example for step-size

We use the same function as in the previous example. Observe that for $(0,0),(0,1),(0,-1)$ we have that the spectral radius of $\nabla^2 f$ is $1,2,2$ respectively (so the minimum of all is 1). We choose $\alpha \geq 2$ and we get that $g(x,y) = (-x, 3y - 2y^3)$. It is not hard to see that gradient descent does not converge (in the first coordinate function g cycles between x and $-x$).

5 Conclusion

Our positive result cannot be improved without making explicit assumptions on the structure of the cost function f nor using beneficial random noise/well chosen initial conditions (as far as convergence is concerned, speed of convergence is not in the scope of this paper). Naturally,

all these directions are of key interest and are the object of recent work (see section 1.1). Keeping up with this simplest, deterministic implementation of gradient descent a natural hypothesis is that (in settings of practical interest) it converges not only to local minimizers but moreover the size of the region of attraction of each local minimizer is in a sense directly proportional to its quality.

Recently, in [20] there has been some progress in proving such statements in non-convex gradient-like systems that arise from learning in games. In such settings, (stable) fixed points correspond to Nash equilibria, but instead of having the typical system performance being dominated by the worst case Nash equilibria (as Price of Anarchy suggests) the regions of attractions of such bad (social) states prove to be minimal and the system works near optimally on average (given uniformly random initial conditions). Extending such statements to actual gradient-dynamics as well as comparing the average case performance of different heuristics even in restricted settings is a fascinating question that could shed more light into the in-many-cases surprising efficiency of the gradient descent method.

Acknowledgements. We are grateful to Prasad Tetali, Jason D. Lee, Max Simchowitz, Michael I. Jordan and Benjamin Recht for their support and helpful discussions and suggestions as well as for the elucidating blog article at `http://www.offconvex.org` on their elegant work, which inspired our investigation. We are also thankful to Nisheeth Vishnoi, on whose blog the article appeared, for pointing it out to us. This work was completed while Ioannis Panageas was a PhD student at Georgia Institute of Technology. Ioannis Panageas would like to acknowledge NSF EAGER award grants CCF-1415496, CCF-1415498 and a MIT-SUTD postdoctoral fellowship. Georgios Piliouras would like to acknowledge SUTD grant SRG ESD 2015 097 and MOE AcRF Tier 2 Grant 2016-T2-1-170. Part of the work was completed while Ioannis Panageas and Georgios Piliouras were visiting scientists at the Simons Institute for the Theory of Computing.

References

1 Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. Simple, efficient, and neural algorithms for sparse coding. In *28th Conference on Learning Theory (COLT)*, pages 113–149, 2015.

2 Emmanuel J Candes, Xiaodong Li, and Mahdi Soltanolkotabi. Phase retrieval via wirtinger flow: Theory and algorithms. *Information Theory, IEEE Transactions on*, 61(4):1985–2007, 2015.

3 Erick Chastain, Adi Livnat, Christos Papadimitriou, and Umesh Vazirani. Algorithms, games, and evolution. *Proceedings of the National Academy of Sciences (PNAS)*, 111(29):10620–10623, 2014.

4 Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. *arXiv preprint arXiv:1412.0233*, 2014.

5 Andrew R Conn, Nicholas IM Gould, and Ph L Toint. *Trust region methods*, volume 1. Siam, 2000.

6 Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in neural information processing systems (NIPS)*, pages 2933–2941, 2014.

7 Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. *arXiv preprint arXiv:1503.02101*, 2015.

8 John L. Kelley. *General Topology*. Springer, 1955.

9 Raghunandan H Keshavan, Sewoong Oh, and Andrea Montanari. Matrix completion from a few entries. *IEEE International Symposium on Information Theory (ISIT)*, pages 324–328, 2009.

10 Robert Kleinberg, Georgios Piliouras, and Eva Tardos. Multiplicative updates outperform generic no-regret learning in congestion games. *Symposium on Theory of Computing (STOC)*, pages 533–542, 2009.

11 Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems (NIPS)*, pages 556–562, 2001.

12 Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. *Conference on Learning Theory (COLT)*, 2016.

13 Ruta Mehta, Ioannis Panageas, and Georgios Piliouras. Natural selection as an inhibitor of genetic diversity: Multiplicative weights updates algorithm and a conjecture of haploid genetics. *Innovations in Theoretical Computer Science (ITCS)*, 2015.

14 Ruta Mehta, Ioannis Panageas, Georgios Piliouras, Prasad Tetali, and Vijay V. Vazirani. Mutation, Sexual Reproduction and Survival in Dynamic Environments. *Innovations in Theoretical Computer Science (ITCS)*, 2017.

15 Ruta Mehta, Ioannis Panageas, Georgios Piliouras, and Sadra Yazdanbod. The Computational Complexity of Genetic Diversity. *European Symposia on Algorithms (ESA)*, 2016.

16 Reshef Meir and David Parkes. On sex, evolution, and the multiplicative weights update algorithm. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 929–937. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

17 Jorge J Moré and Danny C Sorensen. On the use of directions of negative curvature in a modified newton method. *Mathematical Programming*, 16(1):1–20, 1979.

18 Yurii Nesterov. *Introductory lectures on convex optimization*, volume 87. Springer Science and Business Media, 2004.

19 Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

20 Ioannis Panageas and Georgios Piliouras. Average case performance of replicator dynamics in potential games via computing regions of attraction. *17th ACM Conference on Economics and Computation (EC)*, 2016.

21 Robin Pemantle. Nonconvergence to unstable points in urn models and stochastic approximations. *The Annals of Probability*, pages 698–712, 1990.

22 Lawrence Perko. *Differential Equations and Dynamical Systems*. Springer, 3nd. edition, 1991.

23 A Ravindran, Gintaras Victor Reklaitis, and Kenneth Martin Ragsdell. *Engineering optimization: methods and applications*. John Wiley & Sons, 2006.

24 Levent Sagun, Leon Bottou, and Yann LeCun. Singularity of the hessian in deep learning. *arXiv preprint arXiv:1611.07476*, 2016.

25 William H Sandholm. Evolutionary game theory. In *Encyclopedia of Complexity and Systems Science*, pages 3176–3205. Springer, 2009.

26 Michael Shub. *Global Stability of Dynamical Systems*. Springer-Verlag, 1987.

27 Michael Spivak. *Calculus On Manifolds: A Modern Approach To Classical Theorems Of Advanced Calculus*. Addison-Wesley, 1965.

28 Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery over the sphere ii: Recovery by riemannian trust-region method. *arXiv preprint arXiv:1511.04777*, 2015.

29 Yuchen Zhang, Xi Chen, Denny Zhou, and Michael I Jordan. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. *Advances in Neural Information Processing Systems (NIPS)*, pages 1260–1268, 2014.

Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent*

Zeyuan Allen-Zhu[1] **and Lorenzo Orecchia**[2]

1 **Institute for Advanced Study, Princeton, USA**
zeyuan@csail.mit.edu
2 **Boston University, USA**
orecchia@bu.edu

Abstract

First-order methods play a central role in large-scale machine learning. Even though many variations exist, each suited to a particular problem, almost all such methods fundamentally rely on two types of algorithmic steps: gradient descent, which yields primal progress, and mirror descent, which yields dual progress.

We observe that the performances of gradient and mirror descent are complementary, so that faster algorithms can be designed by *linearly coupling* the two. We show how to reconstruct Nesterov's accelerated gradient methods using linear coupling, which gives a cleaner interpretation than Nesterov's original proofs. We also discuss the power of linear coupling by extending it to many other settings that Nesterov's methods cannot apply to.

1998 ACM Subject Classification G.1.6 Optimization, F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases linear coupling, gradient descent, mirror descent, acceleration

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.3

1 Introduction

The study of fast iterative methods for approximately solving convex problems is a central research focus in Machine Learning, Combinatorial Optimizations and many other areas of Computer Science and Mathematics. For large-scale programs, first-order iterative methods are usually the methods of choice due to their cheap and often highly parallelizable iterations.

First-order methods access the target optimization problem $\min_{x\in Q} f(x)$ in a black-box fashion: the algorithm queries a point $y \in Q$ at every iteration and receives the pair $\big(f(y), \nabla f(y)\big)$.[1] The complexity of a first-order method is usually measured in the number of queries necessary to produce an additive ε-approximate minimizer. First-order methods have recently experienced a renaissance in the design of fast algorithms for fundamental computer science problems, varying from discrete ones such as maximum flow problems [20], to continuous ones such as empirical risk minimization [39].

Despite the myriad of applications, first-order methods with provable convergence guarantees can be mostly classified as instantiations of two fundamental algorithmic ideas: *gradient*

* The authors would like to thank Silvio Micali for listening to our work and suggesting the name "linear coupling". The full version of this paper can be found on arXiv https://arxiv.org/abs/1407.1537.

[1] Here, variable x is constrained to lie in a convex set $Q \subseteq \mathbb{R}^n$, which is known as the *constraint set* of the problem.

© Zeyuan Allen-Zhu and Lorenzo Orecchia;
licensed under Creative Commons License CC-BY

descent and the *mirror descent*.[2] We argue that gradient descent takes a fundamentally primal approach, while mirror descent follows a complementary dual approach. In our main result, we show how these two approaches blend in a natural manner to yield a new and simple accelerated gradient method for smooth convex optimization problems, as well as lead to other applications where the classical accelerated gradient methods do not apply.

1.1 Understanding First-Order Methods: Gradient Descent and Mirror Descent

We now provide high-level descriptions of gradient and mirror descent. While this material is classical, our intuitive presentation of these ideas forms the basis for our main result in the subsequent sections. For a more detailed survey, we recommend the textbooks [9, 26].

Consider for simplicity the unconstrained minimization (i.e. $Q = \mathbb{R}^n$), but, as we will see in Section 2, the same intuition and a similar analysis extend to the constrained or even the proximal case. We use generic norms $\|\cdot\|$ and their duals $\|\cdot\|_*$. At a first reading, they can be both replaced with the Euclidean norm $\|\cdot\|_2$.

1.1.1 Primal Approach: Gradient Descent

A natural approach to iterative optimization is to decrease the objective function as much as possible at every iteration. To formalize the effectiveness of this idea, one usually introduces a smoothness assumption on the objective $f(x)$. Specifically, recall that an L-smooth function f satisfies $\|\nabla f(x) - \nabla f(y)\|_* \leq L\|x - y\|$ for every x, y. Such a smoothness condition yields a global quadratic upper bound on the function around a query point x:

$$\forall y, \quad f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|y - x\|^2 \ . \tag{1.1}$$

Gradient-descent algorithms exploit this bound by taking a step that maximizes the guaranteed objective decrease (i.e., the primal progress) $f(x_k) - f(x_{k+1})$ at every iteration k. More precisely,

$$x_{k+1} \leftarrow \arg\min_y \left\{ \frac{L}{2}\|y - x_k\|^2 + \langle \nabla f(x), y - x_k \rangle \right\} \ .$$

Notice that here $\|\cdot\|$ is a generic norm. When this is the Euclidean ℓ_2-norm, the step takes the familiar additive form $x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$. However, in other cases, e.g., for the non-Euclidean ℓ_1 or ℓ_∞ norms, the update step will not follow the direction of the gradient $\nabla f(x_k)$ (see for instance [18, 27]).

Under the smoothness assumption above, the magnitude of this primal progress is at least

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2L}\|\nabla f(x_k)\|_*^2 \ . \tag{1.2}$$

In general, this quantity will be larger when the gradient $\nabla f(x_k)$ has large norm. Classical convergence analysis of gradient descent usually combines (1.2) with a basic convexity argument to relate $f(x_k) - f(x^*)$ and $\|\nabla f(x_k)\|_*$: that is, $f(x_k) - f(x^*) \leq \|\nabla f(x_k)\|_* \|x_k - x^*\|$.

[2] We emphasize here that these two terms are sometimes used ambiguously in the literature; in this paper, we attempt to stick as close as possible to the conventions of the optimization community and in particular in the textbooks [9, 26] with one exception: we extend the definition of gradient descent to non-Euclidean norms in a natural way, following [18].

For L-smooth objectives, the final bound shows that gradient descent converges in $O\left(\frac{L}{\varepsilon}\right)$ iterations [26].

The limitation of gradient descent is that it does not make any attempt to construct a good lower bound to the optimum value $f(x^*)$. It essentially ignores the dual problem. In the next subsection, we review mirror descent, a method that focuses completely on the dual side.

1.1.2 Dual Approach: Mirror Descent

Mirror-descent methods (see for instance [9, 12, 24, 28, 44]) tackle the dual problem by constructing lower bounds to the optimum. Recall that each queried gradient $\nabla f(x)$ can be viewed as a hyperplane lower bounding the objective f: that is, $f(u) \geq f(x) + \langle \nabla f(x), u - x \rangle$ for all u. Mirror-descent methods attempt to carefully construct a convex combination of these hyperplanes in order to yield even a stronger lower bound. Formally, suppose one has queried points $x_0, \ldots, x_{k-1}$, then we form a linear combination of the k hyperplanes and obtain[3]

$$\forall u, \quad f(u) \geq \frac{1}{k}\sum_{t=0}^{k-1} f(x_t) + \frac{1}{k}\sum_{t=0}^{k-1} \langle \nabla f(x_t), u - x_t \rangle \ . \tag{1.3}$$

On the upper bound side, we consider a simple choice $\overline{x} = \frac{1}{k}\sum_{t=0}^{k-1} x_t$, i.e., the mean of the queried points. By straightforward convexity argument, we have $f(\overline{x}) \leq \frac{1}{k}\sum_{t=0}^{k-1} f(x_t)$. As a result, the distance between $f(\overline{x})$ and $f(u)$ for any arbitrary u can be upper bounded using (1.3):

$$\forall u, \quad f(\overline{x}) - f(u) \leq \frac{1}{k}\sum_{t=0}^{k-1} \langle \nabla f(x_t), x_t - u \rangle \stackrel{\text{def}}{=} R_k(u) \ . \tag{1.4}$$

Borrowing terminology from online learning, the right hand side $R_k(u)$ is known as the *regret* of the sequence $(x_t)_{t=0}^{k-1}$ with respect to point u. Now, consider a regularized version $\widetilde{R}_k(u)$ of the regret

$$\widetilde{R}_k(u) \stackrel{\text{def}}{=} \frac{1}{k} \cdot \Big(-\frac{w(u)}{\alpha} + \sum_{t=0}^{k-1} \langle \nabla f(x_t), x_t - u \rangle \Big) \ ,$$

where $\alpha > 0$ is a trade-off parameter and $w(\cdot)$ is some regularizer that is usually strongly convex. Then, mirror-descent methods choose the next iterate x_k by minimizing the maximum regularized regret at the next iteration: that is, choose $x_k \leftarrow \arg\max_u \widetilde{R}_k(u)$. This update rule can be shown to successfully drive $\max_u \widetilde{R}_k(u)$ down as k increases, and thus the right hand side of (1.4) decreases as k increases. This can be made into a rigorous analysis and show that mirror descent converges in $T = O(\rho^2/\varepsilon^2)$ iterations. Here, ρ^2 is the average value of $\|\nabla f(x_k)\|_*^2$ across the iterations.

To sum up, the smaller the queried gradients are (i.e. the smaller $\|\nabla f(x_k)\|_*$ is), the tighter the lower bound (1.3) becomes, and therefore the fewer iterations are needed for mirror descent to converge. (Note that the above mirror-descent analysis can also be used to derive the $1/\varepsilon$ convergence rate on smooth objectives similar to that in gradient descent [11]; since this adaption is not needed in our paper, we omit the details.)

[3] For simplicity, we choose uniform weights here. For the purpose of proving convergence results, the weights of individual hyperplanes are typically uniform or only dependent on k.

Remarks. Mirror descent admits several different algorithmic implementations, such as *Nemirovski's mirror descent* [24] and *Nesterov's dual averaging* [28].[4] Results based on one implementation can usually be transformed into another with some efforts. In this paper, we adopt Nemirovski's mirror descent as our choice of mirror descent, see Section 2.2.

One may occasionally find analyses that do not immediately fall into the above two categories. To name a few, solely using mirror descent and dual lower bounds, one can also obtain a convergence rate $1/\varepsilon$ for smooth objectives similar to that in gradient descent [11]. Conversely, one can deduce the mirror-descent guarantee by applying gradient descent on a dual objective (see Appendix A.3). Shamir and Zhang [40] obtained an algorithm that converges slightly slower than mirror descent, but has an error guarantee on the last iterate, rather than the average history.

1.2 Our Conceptual Question

Following this high level description of gradient and mirror descent, it is useful to pause and observe the complementary nature of the two procedures. Gradient descent relies on primal progress, uses local steps and makes faster progress when the norms of the queried gradients $\|\nabla f(x_k)\|$ are *large*. In contrast, mirror descent works by ensuring dual progress, uses global steps and converges faster when the norms of the queried gradients $\|\nabla f(x_k)\|$ are *small*.

This interpretation immediately leads to the question that inspires our work:

Can Gradient Descent and Mirror Descent be combined to obtain faster first-order methods?

In this paper, we initiate the formal study of this key conceptual question, and propose a *linear coupling* framework. To properly discuss our framework, we choose to mostly focus in the context of convex smooth minimization, and show how to reconstruct Nesterov's accelerated gradient methods using linear coupling. We also discuss the power of our framework by extending it to many other settings beyond Nesterov's original scope.

1.3 Accelerated Gradient Method Via Linear Coupling

In the seminal work [25, 26], Nesterov designed an accelerated gradient method for L-smooth functions with respect to ℓ_2 norms, and it performs quadratically faster than gradient descent – requiring $\Omega(L/\varepsilon)^{0.5}$ rather than $\Omega(L/\varepsilon)$ iterations. This is asymptotically tight [26]. Later in 2005, Nesterov generalizes his method to allow non-Euclidean norms in the definition of smoothness [27]. All these versions of methods are referred to as *accelerated gradient methods*, or sometimes as Nesterov's accelerated methods.

Although accelerated gradient methods have been widely applied (to mention a few, see [38, 39] for regularized optimizations, [19, 30] for composite optimization, [29] for cubic regularization, [31] for universal method, and [20] for an application on maxflow), they are often regarded as "analytical tricks" [17] because their convergence analyses are somewhat complicated and lack of intuitions.

In this paper, we provide a simple, alternative, but **complete** version of the accelerated gradient method. Here, by "complete" we mean our method works for any norm, and for both

[4] Other update rules can be viewed as specializations or generalizations of the mentioned implementations. For instance, the follow-the-regularized-leader (FTRL) step is a generalization of Nesterov's dual averaging step where the regularizers are can be adaptively selected (see [23]).

the constrained and unconstrained case.[5] Our key observation is to construct two sequences of updates: one sequence of gradient-descent updates and one sequence of mirror-descent updates.

Thought Experiment. Consider $f(x)$ that is unconstrained and L-smooth. For sake of demonstrating the idea, suppose $\|\nabla f(x)\|_2$, the norm of the observed gradient, is *either* always $\geq K$, or always $\leq K$, where the cut-off value K is determined later. Under such "wishful assumption", we propose the following algorithm: if $\|\nabla f(x)\|_2$ is always $\geq K$, we perform T gradient-descent steps; otherwise we perform T mirror-descent steps.

To analyze such an algorithm, suppose without loss of generality we start with some point x_0 whose objective distance $f(x_0) - f(x^*)$ is at most 2ε, and we want to find some x so that $f(x) - f(x^*) \leq \varepsilon$.[6] If T gradient-descent steps are performed, the objective decreases by at least $\frac{\|\nabla f(\cdot)\|_2^2}{2L} \geq \frac{K^2}{2L}$ per step according to (1.2), and we only need $T \geq \Omega(\frac{\varepsilon L}{K^2})$ steps to achieve an ε accuracy. If T mirror-descent steps are performed, we need $T \geq \Omega(\frac{K^2}{\varepsilon^2})$ steps according to the mirror-descent convergence. In sum, we need $T \geq \Omega\big(\max\big\{\frac{\varepsilon L}{K^2}, \frac{K^2}{\varepsilon^2}\big\}\big)$ steps to converge to an ε-minimizer. Setting K to be the "magic number" to balance the two terms, we only need $T \geq \Omega\big(\frac{L}{\varepsilon}\big)^{1/2}$ iterations as desired.

Towards an Actual Proof. To turn our thought experiment into an actual proof, we face the following obstacles. Although gradient-descent steps always decrease the objective, mirror-descent steps may sometimes increase the objective, cancelling the effect of the gradient descent. On the other hand, the mirror-descent steps are *only* useful when a large number of iterations are performed in a row; if any gradient-descent step stands in the middle, the convergence is destroyed.

For this reason, it is natural to design an algorithm that, in every single iteration k, performs *both* a gradient and a mirror descent step, and somehow ensure that the two steps are coupled together. However, the following additional difficulty arises: if from some starting point x_k, the gradient-descent step instructs us to go to y_k, while the mirror-descent step instructs us to go to z_k, then how do we continue? Do we look at the gradient at $\nabla f(y_k)$ or $\nabla f(z_k)$ in the next iteration?

This problem is implicitly solved by Nesterov using the following simple idea[7]: in the k-th iteration, we choose a linear combination $x_{k+1} \leftarrow \tau z_k + (1-\tau)y_k$, and use this same gradient $\nabla f(x_{k+1})$ to continue the gradient and mirror steps of the next iteration. Whenever τ is carefully chosen (just like the "magic number" K), the two descent sequences provide a coupled bound on the error guarantee, and we recover the same convergence as [27].

Roadmap. We review the key lemmas of gradient and mirror descent in Section 2. We propose a simple method with fixed step length to recover Nesterov's accelerated methods for the unconstrained case in Section 3, and generalize it to the full-setting in Section 4. We

[5] Some authors have regarded the result in [26] as "momentum analysis" [32, 41] or "ball method" [10]. These analyses only apply to Euclidean spaces. We point out the importance of allowing non-Euclidean norms in Appendix A.1. In addition, our proof in this paper extends naturally to the proximal version of first-order methods, but for simplicity, we choose to include only the constrained version.

[6] For all first-order methods, the heaviest computation always happens in this 2ε to ε process.

[7] We wish to point out that Nesterov has phrased his method differently from ours, and little is known on why this linear combination is needed from his proof, except for being used as an algebraic trick to cancel specific terms.

discuss several important applications of linear coupling that Nesterov's original methods do not solve in Section 5.

2 Key Lemmas of Gradient and Mirror Descent

2.1 Review of Gradient Descent

Consider a function $f(x)$ that is convex and differentiable on a closed convex set $Q \subseteq \mathbb{R}^n$, and assume that f is *L-smooth* with respect to $\|\cdot\|$, that is, for every $x, y \in Q$, it satisfies $\|\nabla f(x) - \nabla f(y)\|_* \le L\|x - y\|$. Here, $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$.[8]

▶ **Definition 2.1.** For any $x \in Q$, the *gradient (descent) step* (with step length $\frac{1}{L}$) is

$$\widetilde{x} = \mathsf{Grad}(x) \stackrel{\text{def}}{=} \arg\min_{y \in Q} \left\{ \frac{L}{2}\|y - x\|^2 + \langle \nabla f(x), y - x \rangle \right\}$$

and we let $\mathsf{Prog}(x) \stackrel{\text{def}}{=} -\min_{y \in Q} \left\{ \frac{L}{2}\|y - x\|^2 + \langle \nabla f(x), y - x \rangle \right\} \ge 0$.

In particular, when $\|\cdot\| = \|\cdot\|_2$ is the ℓ_2-norm and $Q = \mathbb{R}^n$ is unconstrained, the gradient step can be simplified as $\mathsf{Grad}(x) = x - \frac{1}{L}\nabla f(x)$. Or, slightly more generally, when $\|\cdot\| = \|\cdot\|_2$ is the ℓ_2-norm but Q may be constrained, we have $\mathsf{Grad}(x) = x - \frac{1}{L} g_Q(x)$ where $g_Q(x)$ is the gradient mapping of f at x (see Chapter 2.2.3 of [26]).

The classical theory on smooth convex programming gives rise to the following lower bound on the amount of objective decrease (proved in Appendix B for completeness):

$$f(\mathsf{Grad}(x)) \le f(x) - \mathsf{Prog}(x) \tag{2.1}$$

$$\text{or in the special case when } Q = \mathbb{R}^n \qquad f(\mathsf{Grad}(x)) \le f(x) - \frac{1}{2L}\|\nabla f(x)\|_*^2 \ .$$

From the above descent guarantee, one can deduce the convergence rate of gradient descent. For instance, if $\|\cdot\| = \|\cdot\|_2$ is the Euclidean norm, and if gradient step $x_{k+1} = \mathsf{Grad}(x_k)$ is applied T times, we obtain the following convergence guarantee (see [26])

$$f(x_T) - f(x^*) \le O\Big(\frac{L\|x_0 - x^*\|_2^2}{T}\Big) \quad \text{or equivalently} \quad T \ge \Omega\Big(\frac{L\|x_0 - x^*\|_2^2}{\varepsilon}\Big) \Rightarrow f(x_T) - f(x^*) \le \varepsilon \ .$$

Here, x^* is any minimizer of $f(x)$. If $\|\cdot\|$ is a general norm, but $Q = \mathbb{R}^n$ is unconstrained, the above convergent rate becomes $f(x_T) - f(x^*) \le O\big(\frac{LR^2}{T}\big)$, where $R = \max_{x: f(x) \le f(x_0)} \|x - x^*\|$. We provide the proof of this later case in Appendix B because it is less known and we cannot find it in the optimization literature.

Note that, we are unaware of any universal convergence proof for both the general norm and the unconstrained case. As we shall see later in Section 4, this convergence rate can be improved by accelerated gradient methods, even for the general norm $\|\cdot\|$ and the constrained case.

2.2 Review of Mirror Descent

Consider some function $f(x)$ that is convex on a closed convex set $Q \subseteq \mathbb{R}^n$, and assume that f is *ρ-Lipschitz continuous* with respect to norm $\|\cdot\|$, that is, for every $x, y \in Q$, it satisfies $|f(x) - f(y)| \le \rho\|x - y\|$. This is equivalent to saying that f admits a subgradient

[8] $\|\xi\|_* \stackrel{\text{def}}{=} \max\{\langle \xi, x \rangle : \|x\| \le 1\}$. For instance, ℓ_p norm is dual to ℓ_q norm if $\frac{1}{p} + \frac{1}{q} = 1$.

$\partial f(x)$ at every point $x \in Q$, and satisfies $\|\partial f(x)\|_* \le \rho$. (Recall that $\partial f(x) = \nabla f(x)$ if f is differentiable.)

Mirror descent requires one to choose a regularizer (also referred to as a distance generating function):

▶ **Definition 2.2.** We say that $w\colon Q \to \mathbb{R}$ is a *distance generating function (DGF)*, if w is 1-strongly convex with respect to $\|\cdot\|$, or in symbols, $\forall x \in Q \setminus \partial Q$, $\forall y \in Q$: $w(y) \ge w(x) + \langle \nabla w(x), y - x\rangle + \frac{1}{2}\|x-y\|^2$. Accordingly, the *Bregman divergence* is given as

$$V_x(y) \stackrel{\text{def}}{=} w(y) - \langle \nabla w(x), y - x\rangle - w(x) \quad \forall x \in Q \setminus \partial Q, \forall y \in Q \ .$$

The property of DGF ensures that $V_x(x) = 0$ and $V_x(y) \ge \frac{1}{2}\|x-y\|^2 \ge 0$.

Common examples of DGFs include `(i)` $w(y) = \frac{1}{2}\|y\|_2^2$, which is strongly convex with respect to the ℓ_2-norm over every Q, and the corresponding $V_x(y) = \frac{1}{2}\|x-y\|_2^2$, and `(ii)` the entropy function $w(y) = \sum_i y_i \log y_i$, which is strongly convex with respect to the ℓ_1-norm over any $Q \subseteq \Delta \stackrel{\text{def}}{=} \{x \ge 0 : \mathbb{1}^T x = 1\}$. and the corresponding $V_x(y) = \sum_i y_i \log(y_i/x_i) \ge \frac{1}{2}\|x-y\|_1^2$.

▶ **Definition 2.3.** The *mirror (descent) step* with step length α can be described as

$$\widetilde{x} = \mathsf{Mirr}_x(\alpha \cdot \partial f(x)) \quad \text{where} \quad \mathsf{Mirr}_x(\xi) \stackrel{\text{def}}{=} \arg\min_{y \in Q} \left\{ V_x(y) + \langle \xi, y - x\rangle \right\} \ .$$

Mirror descent's core lemma is the following inequality (proved in Appendix B for completeness):

If $x_{k+1} = \mathsf{Mirr}_{x_k}\big(\alpha \cdot \partial f(x_k)\big)$, then

$$\forall u \in Q, \quad \alpha(f(x_k) - f(u)) \le \alpha\langle \partial f(x_k), x_k - u\rangle \le \frac{\alpha^2}{2}\|\partial f(x_k)\|_*^2 + V_{x_k}(u) - V_{x_{k+1}}(u) \ . \tag{2.2}$$

The term $\langle \partial f(x_k), x_k - u\rangle$ features prominently in online optimization, and is known as the *regret* at iteration k with respect to u (see Appendix A.2 for the folklore relationship between mirror descent and regret minimization). It is not hard to see that, telescoping (2.2) for $k = 0, \dots, T-1$, setting $\overline{x} \stackrel{\text{def}}{=} \frac{1}{T}\sum_{k=0}^{T-1} x_k$ to be the average of the x_k's, and choosing $u = x^*$, we have

$$\alpha T(f(\overline{x}) - f(x^*)) \le \sum_{k=0}^{T-1} \alpha\langle \partial f(x_k), x_k - x^*\rangle \le \frac{\alpha^2}{2}\sum_{k=0}^{T-1}\|\partial f(x_k)\|_*^2 + V_{x_0}(x^*) - V_{x_T}(x^*) \ . \tag{2.3}$$

Finally, letting Θ be any upper bound on $V_{x_0}(x^*)$ (recall $\Theta = \frac{1}{2}\|x_0 - x^*\|_2^2$ when $\|\cdot\|$ is the Euclidean norm), and $\alpha = \frac{\sqrt{2\Theta}}{\rho\cdot\sqrt{T}}$ be the step length, inequality (2.2) can be re-written as

$$f(\overline{x}) - f(x^*) \le \frac{\sqrt{2\Theta}\cdot\rho}{\sqrt{T}} \quad \text{or equivalently} \quad T \ge \frac{2\Theta\cdot\rho^2}{\varepsilon^2} \Rightarrow f(\overline{x}) - f(x^*) \le \varepsilon \ . \tag{2.4}$$

Remark. While their analyses share some similarities, mirror and gradient steps are often very different. For example, if the optimization problem is over the simplex with ℓ_1 norm, then gradient step gives $x' \leftarrow \arg\min_y\{\frac{1}{2}\|y-x\|_1^2 + \alpha\langle \nabla f(x), y - x\rangle\}$, while the mirror step with entropy regularizer gives $x' \leftarrow \arg\min_y\{\sum_i y_i \log(y_i/x_i) + \alpha\langle \nabla f(x), y - x\rangle\}$. We point out in Appendix A.1 that non-Euclidean norms are very important for certain applications.

In the special case of $w(x) = \frac{1}{2}\|x\|_2^2$ and $\|\cdot\|$ is ℓ_2-norm, gradient and mirror steps are indistinguishable from each other. However, as we have discussed earlier, these two update rules are often equipped with very different convergence analyses, even if they 'look the same'.

3 Warm-Up Method with Fixed Step Length

Consider the same setting as Section 2.1: that is, $f(x)$ is convex and differentiable on its domain Q, and is L-smooth with respect to some norm $\|\cdot\|$. (Note that $f(x)$ may not have a good Lipschitz continuity parameter ρ, but we do not need such a property.) In this section, we focus on the unconstrained case $Q = \mathbb{R}^n$, and combine gradient and mirror descent to produce a very simple accelerated method. We explain this method first because it avoids the mysterious choices of step lengths as in the full setting, and carries our conceptual message in a very clean way.

Design an algorithm that, in every step k, performs *both* a gradient and a mirror step, and ensures that the two steps are linearly coupled. More specifically, starting from $x_0 = y_0 = z_0$, in each iteration $k = 0, 1, \ldots, T-1$, we first define $x_{k+1} \leftarrow \tau z_k + (1-\tau) y_k$ and then

- perform a gradient step $y_{k+1} \leftarrow \mathsf{Grad}(x_{k+1})$, and
- perform a mirror step $z_{k+1} \leftarrow \mathsf{Mirr}_{z_k}\big(\alpha \nabla f(x_{k+1})\big)$.[9]

Above, α is the (fixed) step length of the mirror step, while τ is the parameter controlling the coupling rate. The choices of α and τ will become clear at the end of this section, but from a high level,

- α will be determined from the mirror-descent analysis, similar to that in (2.3), and
- τ will be determined as the best parameter to balance the gradient and mirror steps, similar to the "magic number" K in our thought experiment discussed in Section 1.3.

Classical gradient-descent and mirror-descent analyses immediately imply the following:

▶ **Lemma 3.1.** *For every $u \in Q = \mathbb{R}^n$,*

$$\begin{aligned} \alpha \langle \nabla f(x_{k+1}), z_k - u \rangle &\overset{①}{\le} \frac{\alpha^2}{2} \|\nabla f(x_{k+1})\|_*^2 + V_{z_k}(u) - V_{z_{k+1}}(u) \\ &\overset{②}{\le} \alpha^2 L \big(f(x_{k+1}) - f(y_{k+1}) \big) + V_{z_k}(u) - V_{z_{k+1}}(u) \ . \end{aligned} \tag{3.1}$$

Proof. To deduce ①, we note that our mirror step $z_{k+1} = \mathsf{Mirr}_{z_k}(\alpha \nabla f(x_{k+1}))$ is essentially identical to that of $x_{k+1} = \mathsf{Mirr}_{x_k}(\alpha \nabla f(x_k))$ in (2.2), with only changes of variable names. Therefore, inequality ① is a simple copy-and-paste from (2.2) after changing the variable names (see the proof of (2.2) for details). The second inequality ② is from the gradient step guarantee $f(x_{k+1}) - f(y_{k+1}) \ge \frac{1}{2L} \|\nabla f(x_{k+1})\|_*^2$ in (2.1). ◀

One can immediately see from Lemma 3.1 that, although the mirror step introduces an error $\frac{\alpha^2}{2} \|\nabla f(x_{k+1})\|_*^2$, this error is proportional to the amount of the gradient-step progress $f(x_{k+1}) - f(y_{k+1})$. This captures the observation we stated in the introduction: if $\|\nabla f(x_{k+1})\|_*$ is large, we can make a large gradient step, or if $\|\nabla f(x_{k+1})\|_*$ is small, the mirror step suffers from a small loss.

If we choose $\tau = 1$ or equivalently $x_{k+1} = z_k$, the left hand side of inequality (3.1) becomes $\langle \nabla f(x_{k+1}), x_{k+1} - u \rangle$, the regret at iteration x_{k+1}. In such a case we wish to telescope it for all iterations k in the spirit of mirror descent (see (2.3)). However, we face the problem that the terms $f(x_{k+1}) - f(y_{k+1})$ do not telescope. [10] On the other hand, if we choose $\tau = 0$ or

[9] Here, the mirror step Mirr is defined by specifying any DGF $w(\cdot)$ that is 1-strongly convex over Q.

[10] In other words, although a gradient step may decrease the objective from $f(x_{k+1})$ to $f(y_{k+1})$, it may also get the objective increased from $f(y_k)$ to $f(x_{k+1})$.

equivalently $x_{k+1} = y_k$, then the terms $f(x_{k+1}) - f(y_{k+1}) = f(y_k) - f(y_{k+1})$ telescope, but the left hand side of (3.1) is no longer the regret. [11]

To overcome this issue, we use linear coupling. We compute and upper bound the difference between the left hand side of (3.1) and the actual "regret":

$$\begin{aligned}&\alpha\langle\nabla f(x_{k+1}), x_{k+1} - u\rangle - \alpha\langle\nabla f(x_{k+1}), z_k - u\rangle \\ &= \alpha\langle\nabla f(x_{k+1}), x_{k+1} - z_k\rangle = \frac{(1-\tau)\alpha}{\tau}\langle\nabla f(x_{k+1}), y_k - x_{k+1}\rangle \le \frac{(1-\tau)\alpha}{\tau}(f(y_k) - f(x_{k+1})). \quad (3.2)\end{aligned}$$

Above, we used the fact that $\tau(x_{k+1} - z_k) = (1-\tau)(y_k - x_{k+1})$, as well as the convexity of $f(\cdot)$. It is now immediate that by choosing $\frac{1-\tau}{\tau} = \alpha L$ and combining (3.1) and (3.2), we have

▶ **Lemma 3.2** (Coupling). *Letting $\tau \in (0,1)$ satisfy that $\frac{1-\tau}{\tau} = \alpha L$, we have that*

$$\forall u \in Q = \mathbb{R}^n, \quad \alpha\langle\nabla f(x_{k+1}), x_{k+1} - u\rangle \le \alpha^2 L\big(f(y_k) - f(y_{k+1})\big) + \big(V_{z_k}(u) - V_{z_{k+1}}(u)\big).$$

It is clear from the above proof that τ is introduced to precisely balance the objective decrease $f(x_{k+1}) - f(y_{k+1})$, and the (possible) objective increase $f(y_k) - f(x_{k+1})$. This is similar to the "magic number" K discussed in the introduction.

Finally Convergence Rate. We telescope inequality Lemma 3.2 for $k = 0, 1, \ldots, T-1$. Setting $\overline{x} \stackrel{\text{def}}{=} \frac{1}{T}\sum_{k=0}^{T-1} x_k$ and $u = x^*$, we have

$$\alpha T(f(\overline{x}) - f(x^*)) \le \sum_{k=0}^{T-1} \alpha\langle\partial f(x_k), x_k - x^*\rangle \le \alpha^2 L\big(f(y_0) - f(y_T)\big) + V_{x_0}(x^*) - V_{x_T}(x^*) \ . \quad (3.3)$$

Suppose our initial point is of error at most d, that is $f(y_0) - f(x^*) \le d$, and suppose $V_{x_0}(x^*) \le \Theta$, then (3.3) gives $f(\overline{x}) - f(x^*) \le \frac{1}{T}\big(\alpha L d + \Theta/\alpha\big)$. Choosing $\alpha = \sqrt{\Theta/Ld}$ to be the value that balances the above two terms,[12] we obtain that $f(\overline{x}) - f(x^*) \le \frac{2\sqrt{L\Theta d}}{T}$. In other words,

$$\text{in } T = 4\sqrt{L\Theta/d} \text{ steps, we can obtain some } \overline{x} \text{ satisfying } f(\overline{x}) - f(x^*) \le d/2,$$

halving the distance to the optimum. If we restart this entire procedure a few number of times, halving the distance for every run, then we obtain an ε-approximate solution in

$$T = O\big(\sqrt{L\Theta/\varepsilon} + \sqrt{L\Theta/2\varepsilon} + \sqrt{L\Theta/4\varepsilon} + \cdots\big) = O\big(\sqrt{L\Theta/\varepsilon}\big)$$

iterations, matching the same running time of Nesterov's accelerated methods [25, 26, 27]. It is important to note here that $\alpha = \sqrt{\Theta/Ld}$ increases as time goes (i.e., as d goes down), and therefore $\tau = \frac{1}{\alpha L + 1}$ decreases as time goes. This lesson instructs us that gradient steps should be given more weights than mirror steps, when it is closer to the optimum.[13]

[11] Indeed, our "thought experiment" in the introduction is conducted *as if* we both had $x_{k+1} = z_k$ and $x_{k+1} = y_k$, and therefore we could arrive at the upcoming (3.3) directly.

[12] This is essentially the same way to choose α in mirror descent, see (2.3).

[13] One may find this counter-intuitive because when it is closer to the optimum, the observed gradients will become smaller, and therefore mirror steps should perform well due to our conceptual message in the introduction. This understanding is incorrect for two reasons. First, when it is closer to the optimum, the threshold between "large" and "small" gradients also become smaller, so one cannot rely only on mirror steps. Second, when it is closer to the optimum, mirror steps are more 'unstable' and may increase the objective more (in comparison to the current distance to the optimum), and thus should be given less weight.

Algorithm 1 $\texttt{AGM}(f, w, x_0, T)$

Input: f a differentiable and convex function on Q that is L-smooth with respect to $\|\cdot\|$; w the DGF function that is 1-strongly convex with respect to the same $\|\cdot\|$ over Q; x_0 some initial point; and T the number of iterations.

Output: y_T such that $f(y_T) - f(x^*) \le \frac{4\Theta L}{T^2}$.

$V_x(y) \stackrel{\text{def}}{=} w(y) - \langle \nabla w(x), y - x\rangle - w(x)$.
$y_0 \leftarrow x_0, \quad z_0 \leftarrow x_0$.
for $k \leftarrow 0$ **to** $T-1$ **do**
  $\alpha_{k+1} \leftarrow \frac{k+2}{2L}$, and $\tau_k \leftarrow \frac{1}{\alpha_{k+1}L} = \frac{2}{k+2}$.
  $x_{k+1} \leftarrow \tau_k z_k + (1-\tau_k) y_k$.
  $y_{k+1} \leftarrow \mathsf{Grad}(x_{k+1})$ $\quad \diamond = \arg\min_{y\in Q}\left\{\frac{L}{2}\|y - x_{k+1}\|^2 + \langle \nabla f(x_{k+1}), y - x_{k+1}\rangle\right\}$
  $z_{k+1} \leftarrow \mathsf{Mirr}_{z_k}\big(\alpha_{k+1}\nabla f(x_{k+1})\big)$ $\quad \diamond = \arg\min_{z\in Q}\left\{V_{z_k}(z) + \langle \alpha_{k+1}\nabla f(x_{k+1}), z - z_k\rangle\right\}$
end for
return y_T.

Conclusion. Equipped with the basic knowledge of gradient descent and mirror descent, the above proof is quite straightforward and gives intuition on how the two "magic numbers" α and τ are selected. However, this simple algorithm has several caveats. First, the value α depends on the knowledge of Θ; second, a good initial distance bound d has to be specified; and third, the algorithm has to be restarted. In the next section, we let α and τ change gradually across iterations. This overcomes the mentioned caveats, and also extends the above analysis to allow Q to be constrained.

4 Final Method with Variable Step Lengths

In this section, we recover the main result of [27] in the constrained case, that is

▶ **Theorem 4.1.** *If $f(x)$ is L-smooth w.r.t. $\|\cdot\|$ on Q, and $w(x)$ is 1-strongly convex w.r.t. $\|\cdot\|$ on Q, then* AGM *outputs y_T satisfying $f(y_T) - f(x^*) \le 4\Theta L/T^2$, where Θ is any upper bound on $V_{x_0}(x^*)$.*

We remark here that it is very important to allow the norm $\|\cdot\|$ to be general, rather than focusing on the ℓ_2-norm as in [26]. See our discussion in Appendix A.1.

Our algorithm AGM (see Algorithm 1) starts from $x_0 = y_0 = z_0$. In each step $k = 0, 1, \ldots, T-1$, it computes $x_{k+1} \leftarrow \tau_k z_k + (1-\tau_k) y_k$ and then

- performs gradient step $y_{k+1} \leftarrow \mathsf{Grad}(x_{k+1})$, and
- performs mirror step $z_{k+1} \leftarrow \mathsf{Mirr}_{z_k}\big(\alpha_{k+1}\nabla f(x_{k+1})\big)$.

Here, α_{k+1} is the step length of mirror descent and will be chosen at the end of this section. The value τ_k is $\frac{1}{\alpha_{k+1}L}$ which is slightly different from $\frac{1}{\alpha L+1}$ used in the warm-up case. (This is necessary to capture the constrained case.) Our choice of α_{k+1} will ensure that $\tau_k \in (0, 1]$ for each k.

Convergence Analysis. We state the analogue of Lemma 3.1 whose proof is in Appendix C:

▶ **Lemma 4.2.** *If $\tau_k = \frac{1}{\alpha_{k+1}L}$, then it satisfies that for every $u \in Q$,*

$$\begin{aligned}\alpha_{k+1}\langle \nabla f(x_{k+1}), z_k - u\rangle &\le \alpha_{k+1}^2 L\,\mathsf{Prog}(x_{k+1}) + V_{z_k}(u) - V_{z_{k+1}}(u)\\ &\le \alpha_{k+1}^2 L\big(f(x_{k+1}) - f(y_{k+1})\big) + V_{z_k}(u) - V_{z_{k+1}}(u)\ .\end{aligned}$$

We state the analogue of Lemma 3.2, whose proof is slightly different and in Appendix C:

▶ **Lemma 4.3** (Coupling). *For any $u \in Q$,*

$$\big(\alpha_{k+1}^2 L\big) f(y_{k+1}) - \big(\alpha_{k+1}^2 L - \alpha_{k+1}\big) f(y_k) + \big(V_{z_{k+1}}(u) - V_{z_k}(u)\big) \le \alpha_{k+1} f(u) \ .$$

We are now ready to prove Theorem 4.1:

Proof of Theorem 4.1. In order to telescope Lemma 4.3, we only need to set the sequence of α_k so that $\alpha_k^2 L \approx \alpha_{k+1}^2 L - \alpha_{k+1}$ as well as $\tau_k = 1/\alpha_{k+1} L \in (0,1]$. In our `AGM`, we let $\alpha_k = \frac{k+1}{2L}$ so that $\alpha_k^2 L = \alpha_{k+1}^2 L - \alpha_{k+1} + \frac{1}{4L}$. Summing up Lemma 4.3 for $k = 0, 1, \dots, T-1$, we obtain

$$\alpha_T^2 L f(y_T) + \sum_{k=1}^{T-1} \frac{1}{4L} f(y_k) + \big(V_{z_T}(u) - V_{z_0}(u)\big) \le \sum_{k=1}^{T} \alpha_k f(u) \ .$$

By choosing $u = x^*$, we notice that $\sum_{k=1}^{T} \alpha_k = \frac{T(T+3)}{4L}$, $f(y_k) \ge f(x^*)$, $V_{z_T}(u) \ge 0$ and $V_{z_0}(x^*) \le \Theta$. Therefore, we obtain

$$\frac{(T+1)^2}{4L^2} L f(y_T) \le \Big(\frac{T(T+3)}{4L} - \frac{T-1}{4L}\Big) f(x^*) + \Theta \ ,$$

which after simplification implies $f(y_T) \le f(x^*) + \frac{4\Theta L}{(T+1)^2}$. ◀

Let us make three remarks.

- `AGM` is slightly different from [27]: (1) we use Nemirovski's mirror steps instead of dual averaging steps, (2) we allow arbitrary starting points x_0, and (3) we use $\tau_k = \frac{2}{k+2}$ rather than $\tau_k = \frac{2}{k+3}$.
- `AGM` is very different from the perhaps better-known version [26], which is known by some authors as the "momentum method" [32, 41]. Momentum methods do not apply to non-Euclidean settings.
- In Appendix D, we also recover the strong convexity version of accelerated gradient methods [26], and thus linear coupling provides a complete proof of all existing accelerated gradient methods.

5 Beyond Accelerated Gradient Methods

Providing an intuitive, yet complete interpretation of accelerated gradient methods is an open question in Optimization [17]. Our result in this paper is one important step towards this general goal. Linear coupling not only gives a reinterpretation of Nesterov's accelerated methods, more importantly, it provides a framework for designing first-order methods in a *bigger agenda*. Since the original version of this paper appeared online, our linear-coupling framework has led to breakthroughs for several problems in computer science. In all such problems, the original Nesterov's accelerated methods do not apply. We illustrate a few examples in this line of research, in order to demonstrate the power and generality of linear coupling.

Recall the key lemmas of gradient and mirror descent in linear coupling (see (3.1)):

$$\text{gradient descent:} \quad f(x_{k+1}) - f(y_{k+1}) \ge \frac{1}{2L} \|\nabla f(x_{k+1})\|_*^2 \tag{5.1}$$

$$\text{mirror descent:} \quad \alpha \langle \nabla f(x_{k+1}), z_k - u \rangle \le \frac{\alpha^2}{2} \|\nabla f(x_{k+1})\|_*^2 + V_{z_k}(u) - V_{z_{k+1}}(u) \tag{5.2}$$

Extension 1: Strengthening (5.2) and (5.1). If f satisfies good properties other than smoothness, one can also develop objective decrease lemma to replace (5.1). In addition, if necessary, a non-strongly convex regularizer can be used in mirror descent to replace (5.2). In either or both such cases, linear coupling can still be used to combine the two methods and obtain faster running times; in contrast, Nesterov's original accelerated methods do not apply.

For example, recent breakthroughs on *positive linear programming (positive LP)* are all based on the above extension of linear coupling [3, 4, 5, 22, 42, 43]. For such LPs, the corresponding objective f is intrinsically non-smooth. Some authors including Nesterov himself have applied simple smoothing to turn f into a smooth variant f', and then minimized f' [27]; however, even if Nesterov's accelerated methods are used to minimize f', the resulting running time scales with the problem's *width*, a parameter that can be exponential in input size.[14] In contrast, if linear coupling is used, one can show that $f(x_{k+1}) - f(y_{k+1})$ is lower bounded by a constant times $\sum_j \max\{|\nabla_j f(x_{k+1})|, 1\}^2$ for the original objective f (see [4]). This is a weaker version of (5.1). However, after linear coupling, it leads to a faster algorithm than naively applying Nesterov's accelerated methods on f' *in all parameter regimes.*

Extension 2: Three-Point Coupling. One may naturally consider linearly coupling for more than two vectors. While this is provably unnecessary for minimizing a smooth objective in the full-gradient setting (because accelerated gradient methods are already optimal), it can be very helpful in the *stochastic-gradient* setting.

More specifically, it was a known obstacle in Nesterov's accelerated methods (including our `AGM`) that if the full gradient $\nabla f(x_{k+1})$ is replaced with a random estimator $\widetilde{\nabla}$ whose expectation $\mathbf{E}[\widetilde{\nabla}] = \nabla f(x_{k+1})$, then acceleration disappears in the worst case. Using linear coupling, we can fix this issue by providing the first direct accelerated *stochastic* gradient method. In [1], the author replaced the coupling step $x_{k+1} \leftarrow \tau z_k + (1-\tau) y_k$ with $x_{k+1} \leftarrow \tau_1 z_k + \tau_2 \widetilde{x} + (1-\tau_2-\tau_1) y_k$, where $\widetilde{x}$ is a snapshot point whose full gradient is computed exactly but very infrequently. Such a "three-point" linear coupling provides an accelerated running time because one can combine (5.1), (5.2), together with a so-called variance-reduction inequality [16] all three at once.

Extension 3: Optimal Sampling Probability. Nesterov's accelerated methods generalize to coordinate-descent settings, that is, to minimize f that is L_i-smooth for each coordinate i. The best known coordinate-descent method [21] samples each coordinate i with probability proportional to L_i, and is based on a randomized version of Nesterov's original analysis. Using linear coupling, the authors of [6] discovered that one should select i with probability proportional to $\sqrt{L_i}$ for an even faster running time.

To illustrate the reasoning behind this, let us revisit (5.1) and (5.2). In the coordinate-descent setting, if we abbreviate x_{k+1} with x, the right hand side of (5.1) simply becomes $\frac{1}{2L_i}(\nabla_i f(x))^2$ if coordinate i is selected. As for (5.2), to ensure its left hand side stays the same in expectation, one should replace $\nabla f(x)$ with $\frac{1}{p_i}\nabla_i f(x)$, where p_i is the probability to select i. As a result, the first term on the right hand side of (5.2) becomes $\frac{\alpha^2}{2p_i^2}(\nabla_i f(x))^2$. By comparing these two new terms $\frac{1}{2L_i}(\nabla_i f(x))^2$ and $\frac{\alpha^2}{2p_i^2}(\nabla_i f(x))^2$, we immediately notice

[14] We recommend interested readers to find detailed discussions in [4] regarding the importance of designing *width-independent* solvers for positive LP. As an illustrative example, in the problem of maximum matching (which can be written as positive LP), the width of the problem is the number of edges in the graph.

that p_i had better be proportional to $\sqrt{L_i}$ in order for the two terms to cancel. This simple idea, fully motivated from linear coupling, leads to the fastest accelerated coordinate-descent method [6].

Extension 4: Supporting Non-Convexity. Consider objectives f that are not even convex but still smooth. For instance, neural network training objectives fall into this class if smoothed activation functions are used. In such a case, both (5.1) and (5.2) remain true. However, when coupling the two steps, we cannot claim $\langle \nabla f(x_{k+1}), x_{k+1} - u \rangle \geq f(x_{k+1}) - f(u)$ because there is no convexity. In [2], the authors discovered that one can use the quadratic lower bound $\langle \nabla f(x_{k+1}), x_{k+1} - u \rangle \geq f(x_{k+1}) - f(u) - \frac{L}{2}\|x_{k+1} - u\|^2$ to replace convexity arguments, and still perform a weaker version of linear coupling. This leads to a stochastic algorithm that converges to approximate saddle-points,[15] outperforming both gradient descent and stochastic gradient descent, the only two known first-order methods with provably convergence guarantees.

Acknowledgements. We thank Jon Kelner and Yin Tat Lee for helpful conversations, and Aaditya Ramdas for pointing out a typo in the previous version of this paper.

This material is based upon work partly supported by the National Science Foundation under Grant CCF-1319460 and by a Simons Graduate Student Award under grant no. 284059.

References

1. Zeyuan Allen-Zhu. Katyusha: Accelerated Variance Reduction for Faster SGD. *ArXiv e-prints*, abs/1603.05953, March 2016.
2. Zeyuan Allen-Zhu and Elad Hazan. Variance Reduction for Faster Non-Convex Optimization. In *ICML*, 2016.
3. Zeyuan Allen-Zhu, Yin Tat Lee, and Lorenzo Orecchia. Using optimization to obtain a width-independent, parallel, simpler, and faster positive SDP solver. In *SODA*, 2016.
4. Zeyuan Allen-Zhu and Lorenzo Orecchia. Nearly-Linear Time Positive LP Solver with Faster Convergence Rate. In *STOC*, 2015.
5. Zeyuan Allen-Zhu and Lorenzo Orecchia. Using optimization to break the epsilon barrier: A faster and simpler width-independent algorithm for solving positive linear programs in parallel. In *SODA*, 2015.
6. Zeyuan Allen-Zhu, Peter Richtárik, Zheng Qu, and Yang Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *ICML*, 2016.
7. Sanjeev Arora, Elad Hazan, and Satyen Kale. Fast Algorithms for Approximate Semidefinite Programming using the Multiplicative Weights Update Method. In *FOCS*, pages 339–348. IEEE, 2005. doi:10.1109/SFCS.2005.35.
8. Sanjeev Arora, Elad Hazan, and Satyen Kale. The Multiplicative Weights Update Method: a Meta-Algorithm and Applications. *Theory of Computing*, 8:121–164, 2012. doi:10.4086/toc.2012.v008a006.
9. Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on Modern Convex Optimization*. Society for Industrial and Applied Mathematics, January 2013. doi:10.1137/1.9780898718829.
10. Sébastien Bubeck, Yin Tat Lee, and Mohit Singh. A geometric alternative to Nesterov's accelerated gradient descent. *ArXiv e-prints*, abs/1506.08187, June 2015. arXiv:abs/1506.08187.

[15] Recall that in general non-convex first-order optimization one can only hope for converging to saddle-points.

11 Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *The Journal of Machine Learning Research*, 13(1):165–202, 2012. arXiv:1012.1367.

12 John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite Objective Mirror Descent. In *COLT*, 2010.

13 Olivier Fercoq and Peter Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015. First appeared on ArXiv 1312.5799 in 2013.

14 Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.

15 Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, August 2007. doi:10.1007/s10994-007-5016-8.

16 Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, NIPS 2013, pages 315–323, 2013.

17 Anatoli Juditsky. Convex optimization ii: Algorithms. Lecture notes, November 2013.

18 Jonathan A. Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. An Almost-Linear-Time Algorithm for Approximate Max Flow in Undirected Graphs, and its Multicommodity Generalizations. In *SODA*, April 2014. doi:10.1137/1.9781611973402.16.

19 Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, January 2011. doi:10.1007/s10107-010-0434-y.

20 Yin Tat Lee, Satish Rao, and Nikhil Srivastava. A new approach to computing maximum flows using electrical flows. In *STOC*, page 755, New York, New York, USA, 2013.

21 Yin Tat Lee and Aaron Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *FOCS*, pages 147–156. IEEE, 2013.

22 Michael W. Mahoney, Satish Rao, Di Wang, and Peng Zhang. Approximating the solution to mixed packing and covering lps in parallel $\widetilde{O}(\epsilon^{-3})$ time. In *ICALP*, 2016.

23 H. Brendan McMahan and Matthew Streeter. Adaptive Bound Optimization for Online Convex Optimization. In *COLT*, 2010. arXiv:1002.4908.

24 Arkadi Nemirovsky and David Yudin. *Problem complexity and method efficiency in optimization.* Nauka Publishers, Moscow (in Russian), 1978. John Wiley, New York (in English) 1983.

25 Yurii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In *Doklady AN SSSR (translated as Soviet Mathematics Doklady)*, volume 269, pages 543–547, 1983.

26 Yurii Nesterov. *Introductory Lectures on Convex Programming Volume: A Basic course*, volume I. Kluwer Academic Publishers, 2004.

27 Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, December 2005. doi:10.1007/s10107-004-0552-5.

28 Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259, June 2007. doi:10.1007/s10107-007-0149-x.

29 Yurii Nesterov. Accelerating the cubic regularization of newton's method on convex problems. *Mathematical Programming*, 112(1):159–181, 2008.

30 Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013. doi:10.1007/s10107-012-0629-5.

31 Yurii Nesterov. Universal gradient methods for convex optimization problems. *Mathematical Programming*, May 2014. doi:10.1007/s10107-014-0790-0.

32 Brendan O'Donoghue and Emmanuel Candès. Adaptive Restart for Accelerated Gradient Schemes. *Foundations of Computational Mathematics*, July 2013. doi:10.1007/s10208-013-9150-3.

33 Lorenzo Orecchia, Sushant Sachdeva, and Nisheeth K. Vishnoi. Approximating the exponential, the lanczos method and an $\widetilde{O}(m)$-time spectral algorithm for balanced separator. In *STOC '12*. ACM Press, November 2012.

34 Serge A. Plotkin, David B. Shmoys, and Éva Tardos. Fast Approximation Algorithms for Fractional Packing and Covering Problems. *Mathematics of Operations Research*, 20(2):257–301, May 1995. doi:10.1287/moor.20.2.257.

35 Ankan Saha, S. V. N. Vishwanathan, and Xinhua Zhang. New Approximation Algorithms for Minimum Enclosing Convex Shapes. In *SODA*, pages 1146–1160, September 2011. arXiv:0909.1062.

36 Shai Shalev-Shwartz. Online Learning and Online Convex Optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012. doi:10.1561/2200000018.

37 Shai Shalev-Shwartz and Yoram Singer. Logarithmic regret algorithms for strongly convex repeated games. Technical report, The Hebrew University, 2007.

38 Shai Shalev-Shwartz and Tong Zhang. Accelerated Mini-Batch Stochastic Dual Coordinate Ascent. In *NIPS*, pages 1–17, May 2013. arXiv:1305.2581.

39 Shai Shalev-Shwartz and Tong Zhang. Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization. In *ICML*, pages 64–72, 2014.

40 Ohad Shamir and Tong Zhang. Stochastic Gradient Descent for Non-smooth Optimization: Convergence Results and Optimal Averaging Schemes. In *Proceedings of the 30th International Conference on Machine Learning - ICML '13*, volume 28, 2013. arXiv:1212.1824.

41 Weijie Su, Stephen Boyd, and Emmanuel Candes. A differential equation for modeling nesterov's accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems*, pages 2510–2518, 2014.

42 Di Wang, Michael W. Mahoney, Nishanth Mohan, and Satish Rao. Faster parallel solver for positive linear programs via dynamically-bucketed selective coordinate descent. *ArXiv e-prints*, abs/1511.06468, November 2015.

43 Di Wang, Satish Rao, and Michael W. Mahoney. Unified acceleration method for packing and covering problems via diameter reduction. In *ICALP*, 2016.

44 Lin Xiao. Dual averaging method for regularized stochastic learning and online optimization. *The Journal of Machine Learning Research*, 11:2543–2596, 2010.

A Several Remarks on First-Order Methods

A.1 Importance of Non-Euclidean Norms

Let us use a simple example to illustrate the importance of allowing arbitrary norms in studying first-order methods.

Consider the saddle point problem of $\min_{x\in\Delta_n}\max_{y\in\Delta_m} y^TAx$, where A is an $m\times n$ matrix, $\Delta_n=\{x\in\mathbb{R}^n : x\geq 0\wedge \mathbb{1}^Tx=1\}$ is the unit simplex in $\mathbb{R}^n$, and $\Delta_m=\{y\in\mathbb{R}^m : y\geq 0\wedge \mathbb{1}^Ty=1\}$. This problem is important to study because it captures packing and covering linear programs that have wide applications in many areas of computer science (see the survey of [8]).

Letting $\mu=\frac{\varepsilon}{2\log m}$, Nesterov has shown that the following objective

$$f_\mu(x)\stackrel{\text{def}}{=}\mu\log\left(\frac{1}{m}\sum_{j=1}^m \exp^{\frac{1}{\mu}(Ax)_j}\right)\ ,$$

when optimized over $x \in \Delta_n$, can yield an additive $\varepsilon/2$ solution to the original saddle point problem [27].

This $f_\mu(x)$ is proven to be $\frac{1}{\mu}$-smooth with respect to the ℓ_1-norm over Δ_n, if all the entries of A are between $[-1, 1]$. Instead, $f_\mu(x)$ is $\frac{1}{\mu}$-smooth with respect to the ℓ_2-norm over Δ_n, *only if* the sum of squares of every row of A is at most 1. This ℓ_2 condition is certainly stronger and less natural than the ℓ_1 condition, and the ℓ_1 condition one leads to the fastest (approximate) width-dependent positive LP solver [27].

Different norm conditions also yield different gradient and mirror descent steps. For instance, in the ℓ_1-norm case, the gradient step is $x' \leftarrow \arg\min_{x' \in \Delta_n} \left\{ \frac{1}{2}\|x' - x\|_1^2 + \alpha\langle \nabla f_\mu(x), x' - x\rangle \right\}$, and the mirror step is $x' \leftarrow \arg\min_{x' \in \Delta_n} \left\{ \sum_{i \in [n]} x'_i \log \frac{x'_i}{x_i} + \alpha\langle \nabla f_\mu(x), x' - x\rangle \right\}$. In the ℓ_2-norm case, gradient and mirror steps are both of the form $x' \leftarrow \arg\min_{x' \in \Delta_n} \left\{ \frac{1}{2}\|x' - x\|_2^2 + \alpha\langle \nabla f_\mu(x), x' - x\rangle \right\}$.

As another example, [35] has shown that the ℓ_1 norm, instead of the ℓ_2 one, is crucial when computing the minimum enclosing ball of points. One can find other applications as well in [27] for the use of non-Euclidean norms, and an interesting example of ℓ_∞-norm gradient descent for nearly-linear time maximum flow in [18].

It is now important to note that, the methods in [25, 26] work only for the ℓ_2-norm case, and it is not clear how the proof can be generalized to other norms until [27]. Some other proofs (such as [13]) only work for the ℓ_2-norm because the mirror steps are described as (a scaled version of) gradient steps.

A.2 Folklore Relationship Between Multiplicative Weight Updates and Mirror Descent

The multiplicative weight update (MWU) method (see the survey of Arora, Hazan and Kale [8]) is a simple method that has been repeatedly discovered in theory of computation, machine learning, optimization, and game theory. The setting of this method is the following.

Let $\Delta_n = \{x \in \mathbb{R}^n : x \geq 0 \wedge \mathbb{1}^T x = 1\}$ be the unit simplex in $\mathbb{R}^n$, and we call any vector in Δ_n an *action*. A player is going to play T actions $x_0, \ldots, x_{T-1} \in \Delta_n$ in a row; only after playing x_k, the player observes a loss vector $\ell_k \in \mathbb{R}^n$ that may depend on x_k, and suffers from a loss value $\langle \ell_k, x_k\rangle$. The MWU method ensures that, if $\|\ell_k\|_\infty \leq \rho$ for all $k \in [T]$, then the player has an (adaptive) strategy to choose the actions such that the average *regret* is bounded:

$$\frac{1}{T}\Big(\sum_{i=0}^{T-1} \langle \ell_k, x_k\rangle - \min_{u \in \Delta_n} \sum_{i=0}^{T-1} \langle \ell_k, u\rangle \Big) \leq O\Big(\frac{\rho\sqrt{\log n}}{\sqrt{T}} \Big) \ . \tag{A.1}$$

The left hand side is called the average regret because it is the (average) difference between the suffered loss $\sum_{i=0}^{T-1} \langle \ell_k, x_k\rangle$, and the loss $\sum_{i=0}^{T-1} \langle \ell_k, u\rangle$ of the best action $u \in \Delta_n$ in hindsight. Another way to interpret (A.1) is to state that we can obtain an average regret of ε using $T = O(\frac{\rho^2 \log n}{\varepsilon^2})$ rounds.

The above result can be proven directly using mirror descent. Letting $w(x) \stackrel{\text{def}}{=} \sum_i x_i \log x_i$ be the entropy DGF over the simplex $Q = \Delta_n$, and its corresponding Bregman divergence $V_x(x') \stackrel{\text{def}}{=} \sum_{i \in [n]} x'_i \log \frac{x'_i}{x_i}$, we consider the following update rule.

Start from $x_0 = (1/n, \ldots, 1/n)$, and update $x_{k+1} = \mathsf{Mirr}_{x_k}\big(\alpha \ell_k\big)$, or equivalently, $x_{k+1,i} = x_{k,i} \cdot \exp^{-\alpha \ell_{k,i}} / Z_k$, where $Z_k > 0$ is the normalization factor that equals to $\sum_{i=1}^n x_{k,i} \cdot$

$\exp^{-\alpha\ell_{k,i}}$.[16] Then, the mirror-descent guarantee (2.2) implies that[17]

$$\forall u \in \Delta_n, \quad \alpha\langle \ell_k, x_k - u\rangle \le \frac{\alpha^2}{2}\|\ell_k\|_\infty^2 + V_{x_k}(u) - V_{x_{k+1}}(u) \ .$$

After telescoping the above inequality for all $k = 0, 1, \ldots, T-1$, and using the upper bounds $\|\ell(x_k)\|_\infty \le \rho$ and $V_{x_0}(u) \le \log n$, we obtain that for all $u \in \Delta_n$,

$$\frac{1}{T}\sum_{k=0}^{T-1}\langle \ell_k, x_k - u\rangle \le \frac{\alpha\rho^2}{2} + \frac{\log n}{\alpha T} \ .$$

Setting $\alpha = \frac{\sqrt{\log n}}{\rho\sqrt{T}}$ we arrive at the desired average regret bound (A.1).

In sum, we have re-deduced the MWU method from mirror descent, and the above proof is quite different from most of the classical analysis of MWU (e.g., [7, 8, 14, 34]). It can be generalized to solve the matrix version of MWU [8, 33], as well as to incorporate the width-reduction technique [8, 34]. We ignore such extensions here because they are outside the scope of this paper.

A.3 Deducing the Mirror-Descent Guarantee via Gradient Descent

In this section, we re-deduce the convergence rate of mirror descent from gradient descent. In particular, we show that the dual averaging steps are equivalent to gradient steps on the Fenchel dual of the regularized regret, and deduce the same convergence bound as (2.4). (Similar proof can also be obtained for mirror steps but is notationally more involved.)

Given a sequence of points $x_0, \ldots, x_{T-1} \in Q$, the (scaled) regret with respect to any point $u \in Q$ is $R(x_0, \ldots, x_{T-1}, u) \stackrel{\text{def}}{=} \sum_{i=0}^{T-1} \alpha\langle \partial f(x_i), x_i - u\rangle$. Since it satisfies that $\alpha T \cdot (f(\overline{x}) - f(u)) \le R(x_0, \ldots, x_{T-1}, u)$, the average regret (after scaling) upper bounds on the distance between any point $f(u)$ and the average $\overline{x} = \frac{1}{T}(x_0 + \cdots + x_{T-1})$. Consider now the regularized regret

$$\widehat{R}(x_0, \ldots, x_{T-1}) \stackrel{\text{def}}{=} \max_{u\in Q}\Big\{ \sum_{i=0}^{T-1} \alpha\langle \partial f(x_i), x_i - u\rangle - w(u)\Big\} \ ,$$

and we can rewrite it using the Fenchel dual $w^*(\lambda) \stackrel{\text{def}}{=} \max_{u\in Q}\{\langle \lambda, u\rangle - w(u)\}$ of $w(\cdot)$:

$$\widehat{R}(x_0, \ldots, x_{T-1}) = w^*\Big(-\alpha \sum_{i=0}^{T-1} \partial f(x_i)\Big) + \sum_{i=0}^{T-1} \alpha\langle \partial f(x_i), x_i\rangle \ .$$

The classical theory of Fenchel duality tells us that $w^*(\lambda)$ is 1-smooth with respect to the dual norm $\|\cdot\|_*$, because $w(\cdot)$ is 1-strongly convex with respect to $\|\cdot\|$. We also have $\nabla w^*(\lambda) = \arg\max_{u\in Q}\{\langle \lambda, u\rangle - w(u)\}$. (See for instance [36].)

With enough notations introduced, let us now minimize $\widehat{R}$ by intelligently selecting $x_0, \ldots, x_{T-1}$. Perhaps a little counter-intuitively, we start from $x_0 = \cdots = x_{T-1} = x^*$ and accordingly $\partial f(x^*) = 0$ (if there are multiple subgradients at x^*, choose the zero one).

[16] This version of the MWU is often known as the Hedge rule [14]. Another commonly used version is to choose $x_{k+1,i} = \frac{x_{k,i}(1-\alpha\ell_{k,i})}{Z_k}$. Since $e^{-t} \approx 1 - t$ whenever $|t|$ is small and our choice of α will make sure that $|\alpha\ell_{k,i}| \ll 1$, this is essentially identical to the Hedge rule.

[17] To be precise, we have replaced $\partial f(x_k)$ with ℓ_k. It is easy to see from the proof of (2.2) that this loss vector ℓ_k does not need to come from the subgradient of some objective $f(\cdot)$.

This corresponds to a regret value of zero and a regularized regret $\widehat{R}(x^*,\dots,x^*) = w^*(0) = -\min_{u\in Q}\{w(u)\}$.

Next, we choose the values of $x_0,\dots,x_{T-1}$ one by one. We choose $x_0 = \arg\min_{u\in Q}\{w(u)\}$ as the starting point.[18] Suppose that the values of $x_0,\dots,x_{k-1}$ are already determined, and we are ready to pick $x_k \in Q$. Let us compute the changes in the regularized regret as a function of x_k:

$$\begin{aligned}\Delta\widehat{R} &= \widehat{R}(x_0,\dots,x_k,x^*,\dots,x^*) - \widehat{R}(x_0,\dots,x_{k-1},x^*,\dots,x^*)\\ &= w^*\Big(-\alpha\sum_{i=0}^{k}\partial f(x_i)\Big) - w^*\Big(-\alpha\sum_{i=0}^{k-1}\partial f(x_i)\Big) + \alpha\langle\partial f(x_k),x_k\rangle\\ &\le \Big\langle\nabla w^*\Big(-\alpha\sum_{i=0}^{k-1}\partial f(x_i)\Big), -\alpha\partial f(x_k)\Big\rangle + \frac{1}{2}\big\|\alpha\partial f(x_k)\big\|_*^2 + \alpha\langle\partial f(x_k),x_k\rangle\ . \qquad \text{(A.2)}\end{aligned}$$

Here, the last inequality is because $w^*(a) - w^*(b) \le \langle\nabla w^*(b), a-b\rangle + \frac{1}{2}\|a-b\|_*^2$, owing to the smoothness of $w^*(\cdot)$. At this moment, it is clear to see that if one chooses

$$x_k = \nabla w^*\Big(-\alpha\sum_{i=0}^{k-1}\partial f(x_i)\Big) = \arg\min_{u\in Q}\Big\{w(u) + \sum_{i=0}^{k-1}\alpha\langle\partial f(x_i),u\rangle\Big\}\ ,$$

the first and third terms in (A.2) cancel out, and we obtain $\Delta\widehat{R} \le \frac{1}{2}\big\|\alpha\partial f(x_k)\big\|_*^2$. In other words, the regularized regret increases by no more than $\frac{1}{2}\big\|\alpha\partial f(x_k)\big\|_*^2 \le \alpha^2\rho^2/2$ in each step, so in the end we have $\widehat{R}(x_0,\dots,x_{T-1}) \le -w(x_0) + \alpha^2\rho^2 T/2$.

In sum, by the definition of the regularized regret, we have

$$\begin{aligned}\alpha T\cdot(f(\overline{x}) - f(x^*)) - w(x^*) &\le \sum_{i=0}^{T-1}\alpha\langle\partial f(x_i), x_i - x^*\rangle - w(x^*)\\ &\le \widehat{R}(x_0,\dots,x_{T-1})\\ &\le -w(x_0) + \frac{\alpha^2\rho^2 T}{2}\ .\end{aligned}$$

This implies the following upper bound on the optimality of $f(\overline{x})$

$$f(\overline{x}) - f(x^*) \le \frac{\alpha\rho^2}{2} + \frac{w(x^*)-w(x_0)}{\alpha T} = \frac{\alpha\rho^2}{2} + \frac{V_{x_0}(x^*)}{\alpha T} \le \frac{\alpha\rho^2}{2} + \frac{\Theta}{\alpha T}\ .$$

Finally, choosing $\alpha = \frac{\sqrt{2\Theta}}{\rho\cdot\sqrt{T}}$ to be the step length, we arrive at $f(\overline{x}) - f(x^*) \le \frac{\sqrt{2\Theta}\cdot\rho}{\sqrt{T}}$, which is the same convergence rate as (2.4).

B Missing Proof of Section 2

For the sake of completeness, we provide self-contained proofs of the mirror descent and mirror descent guarantees in this section.

[18] Dual averaging steps typically demand the first point x_0 to be at the minimum of the regularizer $w(\cdot)$, because that leads to the cleanest analysis. This can be relaxed to allow an arbitrary starting point.

B.1 Missing Proof for Gradient Descent

Gradient Descent Guarantee

$$f(\mathsf{Grad}(x)) \le f(x) - \mathsf{Prog}(x) \qquad (2.1)$$

or in the special case when $Q = \mathbb{R}^n$ $\quad f(\mathsf{Grad}(x)) \le f(x) - \frac{1}{2L}\|\nabla f(x)\|_*^2$.

Proof. [19] Letting $\widetilde{x} = \mathsf{Grad}(x)$, we prove the first inequality by

$$\mathsf{Prog}(x) = -\min_{y\in Q}\Big\{\frac{L}{2}\|y-x\|^2 + \langle \nabla f(x), y-x\rangle\Big\} = -\Big(\frac{L}{2}\|\widetilde{x}-x\|^2 + \langle \nabla f(x), \widetilde{x}-x\rangle\Big)$$
$$= f(x) - \Big(\frac{L}{2}\|\widetilde{x}-x\|^2 + \langle \nabla f(x), \widetilde{x}-x\rangle + f(x)\Big) \le f(x) - f(\widetilde{x}) \ .$$

Here, the last inequality is a consequence of the smoothness assumption: for any $x, y \in Q$,

$$\begin{aligned} f(y) - f(x) &= \int_{\tau=0}^{1} \langle \nabla f(x+\tau(y-x)), y-x\rangle d\tau \\ &= \langle \nabla f(x), y-x\rangle + \int_{\tau=0}^{1} \langle \nabla f(x+\tau(y-x)) - \nabla f(x), y-x\rangle d\tau \\ &\le \langle \nabla f(x), y-x\rangle + \int_{\tau=0}^{1} \|\nabla f(x+\tau(y-x)) - \nabla f(x)\|_* \cdot \|y-x\| d\tau \\ &\le \langle \nabla f(x), y-x\rangle + \int_{\tau=0}^{1} \tau L\|y-x\| \cdot \|y-x\| d\tau \\ &= \langle \nabla f(x), y-x\rangle + \frac{L}{2}\|y-x\|^2 \end{aligned}$$

The second inequality follows because in the special case of $Q = \mathbb{R}^n$, we have

$$\mathsf{Prog}(x) = -\min_{y\in Q}\Big\{\frac{L}{2}\|y-x\|^2 + \langle \nabla f(x), y-x\rangle\Big\} = \frac{1}{2L}\|\nabla f(x)\|_*^2 \ . \qquad \blacktriangleleft$$

▶ **Fact 2.1** (Gradient Descent Convergence). *Let $f(x)$ be a convex, differentiable function that is L-smooth with respect to $\|\cdot\|$ on $Q = \mathbb{R}^n$, and x_0 any initial point in Q. Consider the sequence of T gradient steps $x_{k+1} \leftarrow \mathsf{Grad}(x_k)$, then the last point x_T satisfies that*

$$f(x_T) - f(x^*) \le O\big(\frac{LR^2}{T}\big) \ ,$$

where $R = \max_{x:f(x)\le f(x_0)} \|x - x^*\|$, and x^* is any minimizer of f.

Proof. [20] Recall that we have $f(x_{k+1}) \le f(x_k) - \frac{1}{2L}\|\nabla f(x_k)\|_*^2$ from (2.1). Furthermore, by the convexity of f and Cauchy-Schwarz we have

$$f(x_k) - f(x^*) \le \langle \nabla f(x_k), x_k - x^*\rangle \le \|\nabla f(x_k)\|_* \cdot \|x_k - x^*\| \le R \cdot \|\nabla f(x_k)\|_* \ .$$

Letting $D_k = f(x_k) - f(x^*)$ denote the distance to the optimum at iteration k, we now obtain two relationships $D_k - D_{k+1} \ge \frac{1}{2L}\|\nabla f(x_k)\|_*^2$ as well as $D_k \le R \cdot \|\nabla f(x_k)\|_*$. Combining these two, we get

$$D_k^2 \le 2LR^2(D_k - D_{k+1}) \implies \frac{D_k}{D_{k+1}} \le 2LR^2\Big(\frac{1}{D_{k+1}} - \frac{1}{D_k}\Big) \ .$$

[19] This proof can be found for instance in the textbook [26].
[20] Our proof follows almost directly from [26], but he only uses the Euclidean ℓ_2 norm.

Noticing that $D_k \geq D_{k+1}$ because our objective only decreases at every round, we obtain that $\frac{1}{D_{k+1}} - \frac{1}{D_k} \geq \frac{1}{2LR^2}$. Finally, we conclude that at round T, we must have $\frac{1}{D_T} \geq \frac{T}{2LR^2}$, finishing the proof that $f(x_T) - f(x^*) \leq \frac{2LR^2}{T}$. ◀

B.2 Missing Proof for Mirror Descent

Mirror Descent Guarantee

If $x_{k+1} = \mathsf{Mirr}_{x_k}\big(\alpha \cdot \partial f(x_k)\big)$, then

$$\forall u \in Q, \quad \alpha(f(x_k) - f(u)) \leq \alpha\langle \partial f(x_k), x_k - u\rangle \leq \frac{\alpha^2}{2}\|\partial f(x_k)\|_*^2 + V_{x_k}(u) - V_{x_{k+1}}(u) \ . \quad (2.2)$$

Proof. [21] we compute that

$$\begin{aligned}
\alpha\langle \partial f(x_k), x_k - u\rangle &= \langle \alpha\partial f(x_k), x_k - x_{k+1}\rangle + \langle \alpha\partial f(x_k), x_{k+1} - u\rangle \\
&\overset{①}{\leq} \langle \alpha\partial f(x_k), x_k - x_{k+1}\rangle + \langle -\nabla V_{x_k}(x_{k+1}), x_{k+1} - u\rangle \\
&\overset{②}{=} \langle \alpha\partial f(x_k), x_k - x_{k+1}\rangle + V_{x_k}(u) - V_{x_{k+1}}(u) - V_{x_k}(x_{k+1}) \\
&\overset{③}{\leq} \Big(\langle \alpha\partial f(x_k), x_k - x_{k+1}\rangle - \frac{1}{2}\|x_k - x_{k+1}\|^2\Big) + \big(V_{x_k}(u) - V_{x_{k+1}}(u)\big) \\
&\overset{④}{\leq} \frac{\alpha^2}{2}\|\partial f(x_k)\|_*^2 + \big(V_{x_k}(u) - V_{x_{k+1}}(u)\big)
\end{aligned}$$

Here, ① is due to the minimality of $x_{k+1} = \arg\min_{x\in Q}\{V_{x_k}(x) + \langle \alpha\partial f(x_k), x\rangle\}$, which implies that $\langle \nabla V_{x_k}(x_{k+1}) + \alpha\partial f(x_k), u - x_{k+1}\rangle \geq 0$ for all $u \in Q$. ② is due to the triangle equality of Bregman divergence.[22] ③ is because $V_x(y) \geq \frac{1}{2}\|x - y\|^2$ by the strong convexity of the DGF $w(\cdot)$. ④ is by Cauchy-Schwarz. ◀

C Missing Proofs of Section 4

▶ **Lemma 4.2.** *If $\tau_k = \frac{1}{\alpha_{k+1}L}$, then it satisfies that for every $u \in Q$,*

$$\begin{aligned}
\alpha_{k+1}\langle \nabla f(x_{k+1}), z_k - u\rangle &\overset{①}{\leq} \alpha_{k+1}^2 L\mathsf{Prog}(x_{k+1}) + V_{z_k}(u) - V_{z_{k+1}}(u) \\
&\overset{②}{\leq} \alpha_{k+1}^2 L\big(f(x_{k+1}) - f(y_{k+1})\big) + V_{z_k}(u) - V_{z_{k+1}}(u) \ .
\end{aligned}$$

Proof. The second inequality ② is again from the gradient descent guarantee $f(x_{k+1}) - f(y_{k+1}) \geq \mathsf{Prog}(x_{k+1})$. To prove ①, we first write down the key inequality of mirror-descent

[21] This proof can be found for instance in the textbook [9].

[22] That is,

$$\begin{aligned}
\forall x, y \geq 0, \quad \langle -\nabla V_x(y), y - u\rangle &= \langle \nabla w(x) - \nabla w(y), y - u\rangle \\
&= (w(u) - w(x) \\
&\quad - \langle \nabla w(x), u - x\rangle) - (w(u) - w(y) - \langle \nabla w(y), u - y\rangle) \\
&\quad - (w(y) - w(x) - \langle \nabla w(x), y - x\rangle) \\
&= V_x(u) - V_y(u) - V_x(y) \ .
\end{aligned}$$

analysis (whose proof is identical to that of (2.2))

$$\begin{aligned}\alpha_{k+1}\langle\nabla f(x_{k+1}), z_k - u\rangle &= \langle\alpha_{k+1}\nabla f(x_{k+1}), z_k - z_{k+1}\rangle + \langle\alpha_{k+1}\nabla f(x_{k+1}), z_{k+1} - u\rangle \\ &\overset{①}{\leq} \langle\alpha_{k+1}\nabla f(x_{k+1}), z_k - z_{k+1}\rangle + \langle -\nabla V_{z_k}(z_{k+1}), z_{k+1} - u\rangle \\ &\overset{②}{=} \langle\alpha_{k+1}\nabla f(x_{k+1}), z_k - z_{k+1}\rangle + V_{z_k}(u) - V_{z_{k+1}}(u) - V_{z_k}(z_{k+1}) \\ &\overset{③}{\leq} \Big(\langle\alpha_{k+1}\nabla f(x_{k+1}), z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|^2\Big) \\ &\quad + \big(V_{z_k}(u) - V_{z_{k+1}}(u)\big)\end{aligned}$$

Here, ① is due to the minimality of $z_{k+1} = \arg\min_{z\in Q}\{V_{z_k}(z) + \langle\alpha_{k+1}\nabla f(x_{k+1}), z\rangle\}$, which implies that $\langle\nabla V_{z_k}(z_{k+1}) + \alpha_{k+1}\nabla f(x_{k+1}), u - z_{k+1}\rangle \geq 0$ for all $u \in Q$. ② is due to the triangle equality of Bregman divergence (see Footnote 22 in Appendix B). ③ is because $V_x(y) \geq \frac{1}{2}\|x-y\|^2$ by the strong convexity of the $w(\cdot)$.

If one stops here and uses Cauchy-Shwartz $\langle\alpha_{k+1}\nabla f(x_{k+1}), z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|^2 \leq \frac{\alpha_{k+1}^2}{2}\|\nabla f(x_{k+1})\|_*^2$, he will get the desired inequality in the special case of $Q = \mathbb{R}^n$, because $\mathsf{Prog}(x_{k+1}) = \frac{1}{2L}\|\nabla f(x_{k+1})\|_*^2$ from (2.1).

For the general unconstrained case, we need to use the special choice of $\tau_k = 1/\alpha_{k+1}L$ follows. Letting $v \overset{\text{def}}{=} \tau_k z_{k+1} + (1-\tau_k)y_k \in Q$ so that $x_{k+1} - v = (\tau_k z_k + (1-\tau_k)y_k) - v = \tau_k(z_k - z_{k+1})$, we have

$$\begin{aligned}&\langle\alpha_{k+1}\nabla f(x_{k+1}), z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|^2 \\ &= \langle\frac{\alpha_{k+1}}{\tau_k}\nabla f(x_{k+1}), x_{k+1} - v\rangle - \frac{1}{2\tau_k^2}\|x_{k+1} - v\|^2 \\ &= \alpha_{k+1}^2 L\left(\langle\nabla f(x_{k+1}), x_{k+1} - v\rangle - \frac{L}{2}\|x_{k+1} - v\|^2\right) \leq \alpha_{k+1}^2 L\mathsf{Prog}(x_{k+1})\end{aligned}$$

where the last inequality is from the definition of $\mathsf{Prog}(x_{k+1})$. ◀

▶ **Lemma 4.3** (Coupling). *For any $u \in Q$,*

$$\big(\alpha_{k+1}^2 L\big)f(y_{k+1}) - \big(\alpha_{k+1}^2 L - \alpha_{k+1}\big)f(y_k) + \big(V_{z_{k+1}}(u) - V_{z_k}(u)\big) \leq \alpha_{k+1}f(u)\ .$$

Proof. We deduce the following sequence of inequalities

$$\begin{aligned}&\alpha_{k+1}\big(f(x_{k+1}) - f(u)\big) \\ &\leq \alpha_{k+1}\langle\nabla f(x_{k+1}), x_{k+1} - u\rangle \\ &= \alpha_{k+1}\langle\nabla f(x_{k+1}), x_{k+1} - z_k\rangle + \alpha_{k+1}\langle\nabla f(x_{k+1}), z_k - u\rangle \\ &\overset{①}{=} \frac{(1-\tau_k)\alpha_{k+1}}{\tau_k}\langle\nabla f(x_{k+1}), y_k - x_{k+1}\rangle + \alpha_{k+1}\langle\nabla f(x_{k+1}), z_k - u\rangle \\ &\overset{②}{\leq} \frac{(1-\tau_k)\alpha_{k+1}}{\tau_k}(f(y_k) - f(x_{k+1})) + \alpha_{k+1}\langle\nabla f(x_{k+1}), z_k - u\rangle \\ &\overset{③}{\leq} \frac{(1-\tau_k)\alpha_{k+1}}{\tau_k}(f(y_k) - f(x_{k+1})) + \alpha_{k+1}^2 L\big(f(x_{k+1}) - f(y_{k+1})\big) + V_{z_k}(u) - V_{z_{k+1}}(u) \\ &\overset{④}{=} \big(\alpha_{k+1}^2 L - \alpha_{k+1}\big)f(y_k) - \big(\alpha_{k+1}^2 L\big)f(y_{k+1}) + \alpha_{k+1}f(x_{k+1}) + \big(V_{z_k}(u) - V_{z_{k+1}}(u)\big)\end{aligned}$$

Here, ① uses the choice of x_{k+1} that satisfies $\tau_k(x_{k+1} - z_k) = (1-\tau_k)(y_k - x_{k+1})$; ② is by the convexity of $f(\cdot)$ and $1 - \tau_k \geq 0$; ③ uses Lemma 4.2; and ④ uses the choice of $\tau_k = 1/\alpha_{k+1}L$. ◀

D Strong Convexity Version of Accelerated Gradient Method

When the objective $f(\cdot)$ is both σ-strongly convex and L-smooth with respect to the same norm $\|\cdot\|_2$, another version of accelerated gradient method exists and achieves a $\log(1/\varepsilon)$ convergence rate [26]. We show in this section that, our method $\texttt{AGM}(f, w, x_0, T)$ can be used to recover that strong-convexity accelerated method in one of the two ways. Therefore, the gradient-mirror coupling interpretation behind our paper still applies to the strong-convexity accelerated method.

One way to recover the strong-convexity accelerated method is to replace the use of the mirror-descent analysis on the regret term by its strong-convexity counterpart (also known as logarithmic-regret analysis, see for instance [15, 37]). This would incur some different parameter choices on α_k and τ_k, and results in an algorithm similar to that of [26].

Another, but simpler way is to recursively apply Theorem 4.1. In light of the definition of strong convexity and Theorem 4.1, we have

$$\frac{\sigma}{2}\|y_T - x^*\|_2^2 \le f(y_T) - f(x^*) \le \frac{4\cdot\frac{1}{2}\|x_0 - x^*\|_2^2 \cdot L}{T^2} \ .$$

In particular, in every $T = T_0 \stackrel{\text{def}}{=} \sqrt{8L/\sigma}$ iterations, we can halve the distance $\|y_T - x^*\|_2^2 \le \frac{1}{2}\|x_0 - x^*\|_2^2$. If we repeatedly invoke $\texttt{AGM}(f, w, \cdot, T_0)$ a sequence of ℓ times, each time feeding the initial vector x_0 with the previous output y_{T_0}, then in the last run of the T_0 iterations, we have

$$f(y_{T_0}) - f(x^*) \le \frac{4\cdot\frac{1}{2^\ell}\|x_0 - x^*\|_2^2 \cdot L}{T_0^2} = \frac{1}{2^{\ell+1}}\|x_0 - x^*\|_2^2 \cdot \sigma \ .$$

By choosing $\ell = \log\big(\frac{\|x_0 - x^*\|_2^2 \cdot \sigma}{\varepsilon}\big)$, we conclude that

▶ **Corollary 4.1.** *If $f(\cdot)$ is both σ-strongly convex and L-smooth with respect to $\|\cdot\|_2$, in a total of $T = O\big(\sqrt{\frac{L}{\sigma}} \cdot \log\big(\frac{\|x_0 - x^*\|_2^2 \cdot \sigma}{\varepsilon}\big)\big)$ iterations, we can obtain some x such that $f(x) - f(x^*) \le \varepsilon$.*

This is slightly better than the result $O\big(\sqrt{\frac{L}{\sigma}} \cdot \log\big(\frac{\|x_0 - x^*\|_2^2 \cdot L}{\varepsilon}\big)\big)$ in Theorem 2.2.2 of [26].

We remark here that O'Donoghue and Candès [32] have studied some heuristic adaptive restarting techniques which suggest that the above (and other) restarting version of the accelerated method practically outperforms the original method of Nesterov.

High Dimensional Random Walks and Colorful Expansion*

Tali Kaufman[1] **and David Mass**[2]

1 **Bar-Ilan University, Ramat Gan, Israel**
kaufmant@mit.edu
2 **Bar-Ilan University, Ramat Gan, Israel**
dudimass@gmail.com

Abstract

Random walks on bounded degree expander graphs have numerous applications, both in theoretical and practical computational problems. A key property of these walks is that they converge rapidly to their stationary distribution.

In this work we *define high order random walks*: These are generalizations of random walks on graphs to high dimensional simplicial complexes, which are the high dimensional analogues of graphs. A simplicial complex of dimension d has vertices, edges, triangles, pyramids, up to d-dimensional cells. For any $0 \leq i < d$, a high order random walk on dimension i moves between neighboring i-faces (e.g., edges) of the complex, where two i-faces are considered neighbors if they share a common $(i+1)$-face (e.g., a triangle). The case of $i = 0$ recovers the well studied random walk on graphs.

We provide a *local-to-global criterion* on a complex which implies *rapid convergence of all high order random walks* on it. Specifically, we prove that if the 1-dimensional skeletons of all the links of a complex are spectral expanders, then for *all* $0 \leq i < d$ the high order random walk on dimension i converges rapidly to its stationary distribution.

We derive our result through a new notion of high dimensional combinatorial expansion of complexes which we term *colorful expansion*. This notion is a natural generalization of combinatorial expansion of graphs and is strongly related to the convergence rate of the high order random walks.

We further show an explicit family of *bounded degree* complexes which satisfy this criterion. Specifically, we show that Ramanujan complexes meet this criterion, and thus form an explicit family of bounded degree high dimensional simplicial complexes in which all of the high order random walks converge rapidly to their stationary distribution.

1998 ACM Subject Classification G.2.1 Combinatorics

Keywords and phrases High dimensional expanders, expander graphs, random walks

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.4

1 Introduction

Expander graphs play a fundamental role in computer science, mathematics, physics and more (see e.g. [8] and [12]). In particular, random walks on bounded degree expander graphs have been proven to be very useful in many applications. A key property of these walks is that they converge rapidly to their stationary distribution. This property allows one to

* This work was partially supported by ERC and BSF.

© Tali Kaufman and David Mass;
licensed under Creative Commons License CC-BY

efficiently sample points from some huge space while investing only a few random bits, which is necessary in many theoretical and practical computational problems.

In recent years a generalization of expander graphs to higher dimensions has emerged. The high dimensional analogue of a graph is called a *simplicial complex*. It can be viewed as a hypergraph with a closure property, namely, for any hyperedge in the hypergraph all of its subsets are also in the hypergraph. An hyperedge of size i is called an $(i-1)$-*face* of the complex. Thus, the vertices of the complex are the 0-faces, the edges are the 1-faces, the triangles are the 2-faces, etc. A complex is termed *bounded degree* if the number of faces of any dimension which are incident to any vertex is bounded by a constant, independent of the number of vertices in the complex.

On high dimensional expanders

The study of high dimensional expanders gains a lot of interest recently. They were proven to be a bridge between deep questions in topology (such as topological overlapping, see [6]) and important questions in theoretical computer science (such as property testing and error-correcting codes, see [10]). The interested reader is referred to [13] for a recent survey on high dimensional expanders. The general hope is that high dimensional expanders could be used in places where expander graphs were not good enough.

Expander graphs have had an enormous effect on computer science. However, there are some striking applications in which expanders yield good results but not good enough. For example, by the celebrated result of Sipser and Spielman [19], expanders imply good error-correcting codes. However, the codes obtained by [19] are not locally testable, namely, the very desired property of local testability (which in fact was the motivating goal in the construction of error-correcting codes from expanders, as discussed in Spielman's thesis [20]) is not known to be achieved by codes based on expanders.

Another canonical example where we have seen the limit of expanders is the new proof of the PCP theorem obtained by Dinur [2]. Dinur builds on properties of expanders and achieves a simpler proof of the celebrated PCP theorem. However, the PCP obtained by expanders is not strong enough to yield good hardness of approximation results.

The thesis we try to pursue here is that high dimensional expanders could potentially be used in applications as above, where (one-dimensional) expanders were not strong enough. We expect high dimensional expanders to yield better results than the well studied (one-dimensional) expanders since they are stronger objects: They expand at *all* dimensions, i.e., with respect to vertices, edges, triangles, etc., and not just with respect to vertices as expander graphs.

Dinur's proof of the PCP theorem builds heavily on random walks on expander graphs. In the following we study high dimensional analogues of random walks in high dimensional simplicial complexes.

On high dimensional random walks

In this paper we define high order random walks on simplicial complexes. Our definition generalizes the well studied random walk on graphs to high dimensional simplicial complexes.

For a d-dimensional simplicial complex, we define the high order random walk on it for any dimension $0 \leq i < d$. This walk moves at random between neighboring i-faces (e.g., edges) of the complex, where two i-faces are considered neighbors if they share a common $(i+1)$-face (e.g., a triangle).

The question we address in this paper is the following.

▶ **Question.** *Are there bounded degree high dimensional simplicial complexes in which all of the high order random walks converge rapidly to their stationary distribution.*

In a nutshell, our answer is the following. We provide a local criterion on a complex which implies the rapid convergence of all high order random walks on it. Moreover, we show an explicit family of bounded degree complexes which satisfy this criterion, and hence are bounded degree high dimensional simplicial complexes in which all of the high order random walks converge rapidly to their stationary distribution.

The convergence rate of random walks on graphs is known to be deduced from the spectrum of the graph's adjacency matrix (for more about random walks on graphs see [11]). Similarly, for a d-dimensional simplicial complex and any $0 \leq i < d$ we define the matrix A_i to be the adjacency matrix of dimension i of the complex. This matrix is indexed by the i-faces of the complex, where $A_i(\sigma, \tau)$ indicate if σ and τ are neighbors. The question we address here is how one could construct a bounded degree d-dimensional simplicial complex (for any $d \in \mathbb{N}$) whose *all* adjacency matrices A_i, $0 \leq i < d$, have a concentrated spectrum, and thus its associated high order random walks (on all of its dimensions) converge rapidly to their stationary distribution.

Bounded versus unbounded

It is easy to obtain an unbounded degree graph in which the random walk will converge rapidly to its stationary distribution. This case is less useful as in most applications it is crucial to have a constant bound on the number of neighbors of any vertex in the graph. In a same way, the case of having an unbounded degree complex so the high order random walks on it will converge rapidly to their stationary distribution is less interesting. The complete high dimensional analogue of the most useful random walks on expander graphs are *bounded degree* high dimensional simplicial complexes in which all of the high order random walks converge rapidly to their stationary distribution.

Better than random

In the case of graphs, a random walk on a random bounded degree graph is known to converge rapidly to its stationary distribution. The constructions of explicit expander graphs seem to try and simulate the behavior of these random objects as much as possible, i.e., the philosophy is that a random graph is the best one can get and with explicit constructions we try to get as close as we can to random graphs. In the case of high dimensional simplicial complexes, a high order random walk on a random bounded degree complex *does not* converge rapidly to its stationary distribution. Thus, this phenomenon we study here is very special as it is something we *cannot achieve with a random object*.

Colorful expansion

Our method in asserting the rapid convergence of the high order random walks is as follows. We introduce a new notion of high order combinatorial expansion of complexes which we term colorful expansion: A d-dimensional simplicial complex is said to be a colorful expander if for any $0 \leq i < d$ and any subset S of i-faces there is a large set of *expanding* $(i+1)$-faces, i.e., faces which are hit by S but are not fully covered by S. The term colorful expansion comes from the fact that if we color the faces in S with one color and the faces not in S with another color, then we get many colorful $(i+1)$-faces. This notion is of an independent interest as it is a natural generalization of combinatorial expansion of graphs to higher dimensions, which

is not implied by previously studied notions such as coboundary or cosystolic expansion (see [4]). We show that colorful expansion of a complex implies that *all* of its adjacency matrices A_i, $0 \leq i < d$, have a concentrated spectrum, and hence all of the high order random walks on it converge rapidly to their stationary distribution.

We provide a local-to-global criterion on a complex which implies colorful expansion. For any face σ in the complex, its *link* is the local view of σ, which is obtained by taking all the faces containing σ and removing σ from all of them. We prove that if the 1-dimensional skeleton (i.e., the underlying graph) of every link in the complex is a spectral expander graph, then the entire complex is a colorful expander.

We conclude by showing that Ramanujan complexes satisfy this criterion, and hence are colorful expanders. We then use the explicit construction of Ramanujan complexes from [14] in order to achieve an explicit family of bounded degree complexes in which all of the high order random walks converge rapidly to their stationary distribution.

1.1 Expander graphs and random walks

Let $G = (V, E)$ be an undirected graph. Denote by A the adjacency matrix of G, where $A(u, v)$ equals the number of edges between u and v (we allow multiple edges). The *degree* of a vertex is the number of edges containing it. Throughout this section assume for simplicity that G is k-regular, i.e., the degree of any vertex is k (we deal with the non-regular case at Section 1.5). The *Cheeger constant* of G is defined as

$$h(G) = \min_{\substack{\emptyset \neq S \subset V \\ |S| \leq |V|/2}} \frac{|E(S, \bar{S})|}{k|S|},$$

where $E(S, \bar{S})$ is the set of edges with one endpoint in S and one endpoint in $\bar{S}$.

If $h(G) \geq \epsilon$ for some constant $\epsilon > 0$, then G is said to be an *ϵ-combinatorial expander*. A family of graphs $\{G_i\}_{i \in \mathbb{N}}$ is called a family of ϵ-combinatorial expanders if there exists a constant $\epsilon > 0$ such that $h(G_i) \geq \epsilon$ for every $i \in \mathbb{N}$. Intuitively, it means that the graph is very well connected, and hence the random walk on it is not likely to "get stuck" in a small subset of vertices.

A *random walk* on G can be described by the following process. We start at a random vertex $v_0 \in V$. Then, if after t steps we are at vertex v_t, we choose one of the edges containing v_t uniformly at random and move to its other endpoint. Thus, for any $u, v \in V$

$$\Pr[v_{t+1} = v \mid v_t = u] = \frac{A(u, v)}{k}.$$

Denote by $\pi_0 \in \mathbb{R}^V$ the probability distribution from which v_0 is chosen. Then the random walk on G yields a sequence of probability distributions $\pi_0, \pi_1, \ldots \in \mathbb{R}^V$. The transition from π_t to π_{t+1} is given by $\pi_{t+1} = \pi_t \widetilde{A}$, where $\widetilde{A} = (1/k)A$ is the *normalized adjacency matrix* of G. Note that $\widetilde{A}(u, v)$ is the probability to move from u to v in a single step, and $\widetilde{A}^t(u, v)$ is the probability that a walk starting at u will be at v after t steps. It follows that for any $t \in \mathbb{N}$, $\pi_t = \pi_0 \widetilde{A}^t$.

Denote by $\mathbf{u} = (1/|V|, \ldots, 1/|V|)$ the uniform distribution on the vertices. If G is connected and non-bipartite, then $\pi_0 \widetilde{A}^t \to \mathbf{u}$ as $t \to \infty$ for any initial probability distribution $\pi_0 \in \mathbb{R}^V$. The random walk on G is said to be *μ-rapidly mixing*, $0 < \mu < 1$, if for any initial probability distribution $\pi_0 \in \mathbb{R}^V$ and any $t \in \mathbb{N}$

$$\|\pi_t - \mathbf{u}\|_2 \leq \mu^t.$$

The common way to deduce the mixing rate of a random walk is by the spectrum of the normalized adjacency matrix of G. Denote by $1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{|V|} \geq -1$ the eigenvalues of $\widetilde{A}$. Then the random walk on G is μ-rapidly mixing for $\mu \leq \max\{|\lambda_2|, |\lambda_{|V|}|\}$. (A proof for this assertion can be found in many places, see [8] for instance.)

Here comes the relation between expanders and rapid mixing of random walks. Since combinatorial expanders are very strongly connected, we expect that random walks on them would mix rapidly. This is indeed true and it is given formally by Cheeger's inequality [1] which states that

$$\lambda_2 \leq 1 - \frac{h(G)^2}{2}.$$

Thus, it suffices to show that a graph is an ϵ-combinatorial expander for some constant $\epsilon > 0$ in order to deduce that the random walk on it converges rapidly to the uniform distribution.

▶ Remark. The cautious reader would note that combinatorial expansion gives us a bound only on λ_2, but for rapid mixing we need to bound $\lambda_{|V|}$ as well. This is not a problem since $\lambda_{|V|} = -1$ if and only if the graph is bipartite. Thus, in most applications it is possible just to add self-loops at each vertex which "destroys the bipartiteness" of the graph and does not slow down the convergence rate too much. Nevertheless, we show at section 1.6 that when dealing with high order random walks the smallest eigenvalue of the normalized adjacency matrix is bounded away from -1. In any case, combinatorial expansion is the crucial condition for rapid mixing of random walks.

1.2 High dimensional simplicial complexes

A *simplicial complex* X with a set of vertices V is a collection of subsets of V with a closure property, namely, for any subset $\sigma \in X$, all of its subsets $\tau \subset \sigma$ are also in X. Each such subset is called a *face* of the complex. The *dimension* of a face $\sigma \in X$ is one less than the number of vertices in it, i.e., $\dim(\sigma) = |\sigma| - 1$. We also define the empty set as a face of the complex, which is the unique face of dimension -1. The dimension of the entire complex is defined as the maximal dimension of a face in it. The complex is said to be *pure* if for any face $\sigma \in X$ with $\dim(\sigma) < \dim(X)$ there is a face of maximal dimension $\tau \in X$, $\dim(\tau) = \dim(X)$, such that $\sigma \subset \tau$. We only deal with pure simplicial complexes.

We denote by $X(i)$ the faces of the complex of dimension i, which are called in short the i-faces of the complex. In the terminology of simplicial complexes, an *i-cochain* is a function $W : X(i) \to \{0, 1\}$. For us it is convenient to view an i-cochain as defining a subset of i-faces, where $\sigma \in W \Leftrightarrow W(\sigma) = 1$. The space of all the i-cochains is denoted by $C^i(X)$.

The *degree* of a face $\sigma \in X$ is the number of faces of maximal dimension which contain σ. Again, we assume for simplicity that the complex is regular, i.e., for any $0 \leq i < d$ and any two i-faces $\sigma, \tau \in X(i)$ it holds that $\deg(\sigma) = \deg(\tau)$ (we deal with non-regularity at Section 1.5). For any i-cochain $W \in C^i(X)$ we define its norm to be the fraction of i-faces which are in W (this definition will be refined at Section 1.5), i.e.,

$$\|W\| = \frac{|W|}{|X(i)|}.$$

The *link* of a face $\sigma \in X$, denoted by X_σ, is the complex obtained by taking all the faces which contain σ, and removing σ from all of them. Intuitively, it is the local view of X "from the eyes of σ". Formally it is given by $X_\sigma = \{\tau \setminus \sigma \mid \tau \in X, \tau \supseteq \sigma\}$. If X is of dimension d, then X_σ is a complex of dimension $d - |\sigma|$. We denote by $\|\cdot\|_\sigma$ the norm associated with the link of σ.

Let us demonstrate the above definitions with a simple example.

▶ **Example 1.1.** Let X be a 2-dimensional simplicial complex. The complex contains faces of dimensions $0, 1$ and 2, which can be viewed as vertices, edges and triangles. $X(0)$ denotes the vertices, $X(1)$ the edges, and $X(2)$ the triangles. The degree of any vertex $u \in X(0)$ is the number of triangles containing it, i.e., $\deg(u) = |\{t \in X(2) \mid u \in t\}|$. The link of a vertex $u \in X(0)$ is a 1-dimensional simplicial complex defined as follows. Any edge of the form $\{u, v\} \in X(1)$ becomes a vertex in the link, i.e., $v \in X_u(0)$, and any triangle of the form $\{u, v, w\} \in X(2)$ becomes an edge in the link, i.e., $\{v, w\} \in X_u(1)$. Thus, the link of u contains vertices and edges, which correspond to edges and triangles of the complex, respectively.

1.3 Our contribution

In this work we introduce a new notion of random walks on high dimensional simplicial complexes. Our main result is a local criterion on a complex which implies the rapid mixing of all of its associated high order random walks. In order to assert their rapid mixing we introduce a new definition for combinatorial expansion of high dimensional simplicial complexes, which we term colorful expansion. We prove that this local criterion on a complex implies colorful expansion, which in turn implies the rapid convergence of all high order random walks on it. We then show an explicit construction of bounded degree complexes which satisfy this criterion, and hence are bounded degree complexes in which all of the high order random walks converge rapidly to their stationary distribution.

1.3.1 High order random walks

Let X be a d-dimensional simplicial complex. We generalize the notion of random walks on graphs to higher dimensions by letting the walk to move on any dimension of the complex. For any $0 \leq i < d$ we define the walk on dimension i of the complex as follows. We start from an initial i-face $\sigma_0 \in X(i)$. Then, if after t steps we are at σ_t, the process of choosing σ_{t+1} can be described by the following two steps:

1. Choose an $(i+1)$-face $\tau \supset \sigma_t$ uniformly at random.
2. Choose an i-face $\sigma_{t+1} \subset \tau$, $\sigma_{t+1} \neq \sigma_t$, uniformly at random and move to it.

We say that $\sigma, \sigma' \in X(i)$ are neighbors, denoted by $\sigma \sim \sigma'$, if they share a common $(i+1)$-face, i.e., $\sigma \cup \sigma' \in X(i+1)$. In order to analyze this walk, we define the following auxiliary graphs.

▶ **Definition 1.2.** Let X be a d-dimensional simplicial complex. For any $0 \leq i < d$ the *i-graph* $G_i = (V_i, E_i)$ (with adjacency matrix A_i) is defined as follows.

- The vertices of G_i are the i-faces of X, i.e., $V_i = X(i)$.
- For any two i-faces $\sigma, \sigma' \in X(i)$ such that $\sigma \sim \sigma'$, there is an edge between the corresponding vertices in G_i, i.e., $E_i = \left\{\{\sigma, \sigma'\} \in \binom{X(i)}{2} \mid \sigma \sim \sigma'\right\}$.

Denote by $\pi_0 \in \mathbb{R}^{X(i)}$ the probability distribution from which σ_0 is chosen. As in graphs, the random walk on dimension i of X yields a sequence of probability distributions $\pi_0, \pi_1, \ldots, \in \mathbb{R}^{X(i)}$. Note that the transition from π_t to π_{t+1} is given by $\pi_{t+1} = \pi_t \widetilde{A}_i$, where $\widetilde{A}_i$ is the normalized adjacency matrix of G_i. In other words, the random walk on dimension i of the complex is equivalent to a random walk on the i-graph as above. Thus, our goal is to construct a complex X such that the random walks on *all* of its induced i-graphs, $0 \leq i < d$, will converge rapidly to the uniform distribution.

1.3.2 Rapid mixing criterion

Recall that the link of any face $\sigma \in X$ is a smaller complex which corresponds to the local view of σ. We essentially show that if every link is expanding in some sense, then all of the high order random walks on X converge rapidly to their stationary distribution.

For any face $\sigma \in X$ the 1-dimensional skeleton of X_σ is defined as $X_\sigma(0) \cup X_\sigma(1)$, i.e., the underlying graph of X_σ. We say that X is a *skeleton expander* if the 1-dimensional skeleton of every link behaves pseudorandomly.

▶ **Definition 1.3** (Skeleton expansion)**.** Let X be a d-dimensional simplicial complex. X is called an *α-skeleton expander*, $\alpha > 0$, if for any face $\sigma \in X$ (including $\sigma = \emptyset$) and any subset of vertices $S \subseteq X_\sigma(0)$ it holds that

$$\|E(S)\|_\sigma \leq \|S\|_\sigma(\|S\|_\sigma + \alpha),$$

where $E(S) \subseteq X_\sigma(1)$ denotes the set of edges with both endpoints in S.

We note that in the above definition we can replace the requirement of α-skeleton expansion for $\sigma = \emptyset$ with the weaker requirement that the complex is connected. This could be done since by [16] if the complex is connected, then the α-skeleton expansion of $\sigma = \emptyset$ follows from the α-skeleton expansion of all $\sigma \in X$, $\sigma \neq \emptyset$.

For some intuition regrading the above definition, consider a random graph $G = (V, E)$ whose edges are distributed uniformly. Let $S \subseteq V$ be any subset of vertices of size $\gamma|V|$. Then the probability for any edge to have both endpoints in S is γ^2, and hence the expected size of $E(S)$ is $\gamma^2|E|$. From this point of view, the α-skeleton expansion property means that the underlying graph of every link "looks like" a random graph, up to an error of α.

The main result of this paper is the following theorem.

▶ **Theorem 1.4** (Rapid mixing criterion, informal, for formal see Theorem 3.2)**.** *If a complex is an α-skeleton expander for small enough α, then all of the high order random walks on it converge rapidly to their stationary distribution.*

1.3.3 Colorful expansion

Our new definition for combinatorial expansion of high dimensional simplicial complexes is the main building block in the rapid mixing proof. For an i-cochain $W \in C^i(X)$, we say that an $(i+1)$-face is *expanding* if it hits W but not fully covered by W.

▶ **Definition 1.5** (Expanding faces)**.** Let X be a d-dimensional simplicial complex. For any i-cochain $W \in C^i(X)$, $0 \leq i < d$, the *expanding faces* of W are defined as

$$\psi(W) = \{\tau \in X(i+1) \mid \exists\, \sigma, \sigma' \subset \tau : \sigma \in W, \sigma' \in X(i) \setminus W\}.$$

Then we define colorful expansion as follows.

▶ **Definition 1.6** (Colorful expander)**.** Let X be a d-dimensional simplicial complex. X is called an *ϵ-colorful expander*, $\epsilon > 0$, if for any i-cochain $W \in C^i(X)$, $0 \leq i < d$, $0 < \|W\| \leq 1/2$,

$$\frac{\|\psi(W)\|}{\|W\|} \geq \epsilon.$$

Our main result is then a corollary of the following two theorems.

▶ **Theorem 1.7** (Colorful expansion criterion, informal, for formal see Theorem 2.1)**.** *If a complex is an α-skeleton expander for small enough α, then it is a colorful expander.*

▶ **Theorem 1.8** (Colorful expansion implies rapid mixing, informal, for formal see Theorem 3.1)**.** *If a complex is a colorful expander, then all of the high order random walks on it converge rapidly to their stationary distribution.*

Since Theorem 1.7 is the main technical part of this paper, we provide here a short overview of the proof.

Informal overview of the proof of Theorem 1.7

We need to show that for any dimension i and any non-empty i-cochain with norm up to $1/2$ there are many expanding faces.

Consider an arbitrary i-cochain W. From a very high level point of view, the proof can be divided into three steps:

1. Mark "selected" faces in dimensions $i, i-1, \ldots, -1$.
2. Find a "good" dimension $j \leq i$.
3. "Climb up" from dimension j to deduce many expanding faces in dimension $i+1$.

In more details, we start by marking all of the i-faces which are in W. Then we mark $(i-1)$-faces, where the rule for marking a face is if many (above some threshold) of the i-faces which contain it are marked. We continue this way down to the lowest dimension of the complex, where each dimension is marked with regard to one dimension above. We term these marked faces as *fat faces* since each marked face contains in its link many i-faces which are in W.

The next step is to look for the highest dimension $j \leq i$ with the property that many fat j-faces contain many non-fat $(j-1)$-faces. By setting a large enough threshold for marking faces we can make sure that the empty set is never fat, which implies that such dimension must exist (Proposition 2.6).

The last step is to deduce from the links of the non-fat $(j-1)$-faces that there exist many expanding $(i+1)$-faces. We already know, by the existence of many fat j-faces in these links, that a large part of W is seen in them (Proposition 2.7). It is left to show that many of the $(i+1)$-faces in these links contain at least one i-face which is not in W (Proposition 2.8).

For that we define *full faces* for any dimension of the complex, where a k-face is full if it contains only fat $(k-1)$-faces. By this definition it follows that a full k-face is seen in the link of any $(k-2)$-face as an edge between two fat vertices. So when considering the link of a non-fat $(k-2)$-face, which by definition contains only a few fat vertices, by the skeleton expansion it contains only a few full k-faces. Thus, the number of the full $(j+1)$-faces in the links of the non-fat $(j-1)$-faces is very small. Then again, by the skeleton expansion and the maximality of j we can deduce that for any $k > j+1$ most of the full k-faces contain only full $(k-1)$-faces. This implies that the fraction of full $(i+1)$-faces is about the same as the fraction of the full $(j+1)$-faces. This actually finishes the proof, since we get many $(i+1)$-faces that are not full, i.e., contain at least one i-face which is not in W.

1.3.4 Explicit construction

As proven in [4], Ramanujan complexes are excellent skeleton expanders. Since our definition of skeleton expansion is slightly different than the one appearing in [4], we modify their proof so the requirement of our definition follows. We show that Ramanujan complexes with thickness large enough are α-skeleton expanders with α as small as we want. Then by the explicit construction of Ramanujan complexes from [14] we achieve the following theorem.

▶ **Theorem 1.9** (Explicit construction, informal, for formal see Theorem 4.1)**.** *There exists an explicit family of bounded degree high dimensional simplicial complexes in which all of the high order random walks converge rapidly to their stationary distribution.*

1.4 Related work

In a recent work Oppenheim [16] has shown that if all the links of a d-dimensional simplicial complex are good spectral expanders and the complex is connected, then the 1-dimensional skeleton of the complex is an expander graph, or in other words, that the graph G_0 is a spectral expander. In this paper we show that the spectral expansion of the links, including the spectral expansion of the 1-dimensional skeleton of the complex, imply the spectral expansion of G_i for *all* $0 < i < d$.

Parzanchevski and Rosenthal in [17] study a different notion of high order random walk. The high order random walk of [17] is designed to expose the topological properties of the complex. Parzanchevski and Rosenthal define a variant of the stationary distribution of the random walk and show its relation to the spectrum of the high order laplacian on the space that is *orthogonal to the coboundaries*. In our work, the stationary distribution of the random walks is already known. Moreover, it is already known that the convergence rate is controlled by the spectrum of the high order normalized adjacency matrices on the space that is *orthogonal to the constant functions*. We show that the expansion of the links implies the concentration of the spectrum of the high order normalized adjacency matrices, a thing that cannot be deduced in any way from [17].

We study the spectrum of the high order normalized adjacency matrices of the complex and derive some good bounds on it from the spectrum of the links. Garland in a seminal work [5] has studied the spectrum of high order laplacians associated with a d-dimensional simplicial complex. Garland has shown that if all the links of a complex are good spectral expanders, then the eigenspace of the weighted oriented laplacian that is orthogonal to the coboundaries has a concentrated spectrum. This is somewhat in the spirit of what we get here. However, Garland could only obtain bounds on the eigenspace orthogonal to the coboundaries, while here we want to get a bound on all the eigenvalues besides the first trivial one corresponding to the constant functions. There is no known way to obtain our result from Garland's argument. See also [7] for more discussion on Garland's work and the fact that it does not imply the required spectrum bound for the eigenvalues on the space that is orthogonal to the constant functions.

In a recent work of the first coauthor and Evra [4] a different notion of high order expansion has been studied, which is called cosystolic expansion. It was shown in [4] that if all the links of a d-dimensional simplicial complex are good spectral expanders and good coboundary expanders, then the $(d-1)$-skeleton of the complex is a cosystolic expander. Though the spirit of the proof there might resemble at first glance the method of the proof that we use here, the obstacles and the solutions are different. They could only show cosystolic expansion of *small* sets. Then they use a reduction of [9] showing that cosystolic expansion of small sets implies cosystolic expansion of the $(d-1)$-skeleton of a given d-dimensional simplicial complex. In our work it is crucial for us to obtain expansion of large sets, whose norm is up to $1/2$, and the reduction of [9] could not work here since it does not imply colorful expansion. Thus, the expansion here is achieved in a method which is different than the one used in [4].

1.5 Dealing with non-regularity

Up to now we assumed that the complexes are regular. In this section we describe the necessary modifications for the general case, where we are not guaranteed to have regularity. We start with non-regular graphs (for more about random walks on non-regular graphs see [18]).

Let $G = (V, E)$ be an undirected graph. Instead of using the Cheeger constant we use the *conductance*, which is its generalized version. For any subset of vertices $S \subseteq V$, its *volume* is defined as $\mathrm{vol}(S) = \sum_{v \in S} \deg(v)$. Then the conductance of any subset of vertices $S \subseteq V$ is defined as

$$\Phi(S) = \frac{|E(S, \bar{S})|}{\mathrm{vol}(S)},$$

and the conductance of the graph is defined as

$$\Phi(G) = \min_{\substack{\emptyset \neq S \subseteq V \\ \mathrm{vol}(S) \leq \mathrm{vol}(V)/2}} \Phi(S).$$

When considering a random walk on G, then for any $u, v \in V$ and any $t \in \mathbb{N}$

$$\Pr[v_{t+1} = v \mid v_t = u] = \frac{A(u, v)}{\deg(u)}.$$

So the transition from π_t to π_{t+1} is given by $\pi_{t+1} = \pi_t(DA)$, where D is the diagonal matrix defined by

$$D(u, v) = \begin{cases} \dfrac{1}{\deg(u)} & \text{if } u = v, \\ 0 & \text{otherwise.} \end{cases}$$

The *stationary distribution* of the walk is the probability distribution $\pi \in \mathbb{R}^V$ for which $\pi(DA) = \pi$. When G is connected and non-bipartite, then for any $v \in V$

$$\pi(v) = \frac{\deg(v)}{\sum_{u \in V} \deg(u)},$$

and for any initial probability distribution $\pi_0 \in \mathbb{R}^V$ it holds that $\pi_0(DA)^t \to \pi$ as $t \to \infty$. The random walk is said to be *μ-rapidly mixing*, $0 < \mu < 1$, if for any initial probability distribution $\pi_0 \in \mathbb{R}^V$ and any $t \in \mathbb{N}$

$$\|\pi_t - \pi\|_2 \leq \sqrt{\frac{d_{max}}{d_{min}}} \mu^t,$$

where $d_{max} = \max_{v \in V}\{\deg(v)\}$ and $d_{min} = \min_{v \in V}\{\deg(v)\}$.

The mixing rate of the random walk on G can be deduced from the spectrum of its normalized adjacency matrix, which is defined in the general case as $\widetilde{A} = D^{1/2} A D^{1/2}$. The largest eigenvalue of $\widetilde{A}$ satisfies $\lambda_1 = 1$ (with corresponding eigenvector $\left(\sqrt{\pi(v)}\right)_{v \in V}$) and the smallest eigenvalue satisfies $\lambda_{|V|} \geq -1$. Similar to the regular case, the random walk on G is μ-rapidly mixing for $\mu \leq \max\{|\lambda_2|, |\lambda_{|V|}|\}$. (We prove this assertion in the appendix.)

The following lemma of Sinclair and Jerrum [18] generalizes Cheeger's inequality to non-regular graphs.

▶ **Lemma 1.10.** *Let $G = (V, E)$ be an undirected graph where multiple edges are allowed, $\widetilde{A}$ its normalized adjacency matrix and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{|V|}$ the eigenvalues of $\widetilde{A}$. Then*

$$\lambda_2 \leq 1 - \frac{\Phi(G)^2}{2}.$$

We describe now the modifications required for non-regular complexes. Let X be a d-dimensional simplicial complex. For any i-cochain $W \in C^i(X)$ we refine the definition of the norm to the general case as

$$\|W\| = \frac{\sum_{\sigma \in W} deg(\sigma)}{\sum_{\tau \in X(i)} deg(\tau)}.$$

An alternative view of this norm, which is very useful for further calculations, is the following. Let $P_d \in X(d)$ be a uniformly random d-face of the complex. For any $i = d-1, \ldots, -1$, let $P_i \in X(i)$ be a random i-face which is obtained by removing a uniformly random vertex from P_{i+1}. Then we get a sequence of random variables $\{P_i\}_{i=-1}^{d}$ such that for any i-cochain $W \in C^i(X)$

$$\|W\| = \Pr[P_i \in W].$$

Now when considering a random walk on dimension i of X, we take into consideration the degrees of the faces. Assuming the walk is now at σ_t, then the process of choosing σ_{t+1} is given as follows:

1. Choose an $(i+1)$-face $\tau \supset \sigma_t$ with probability *proportional to its degree.*
2. Choose an i-face $\sigma_{t+1} \subset \tau$, $\sigma_{t+1} \neq \sigma_t$, uniformly at random and move to it.

So the exact probability for moving from σ to σ', where $\sigma \sim \sigma'$, is given by

$$\Pr[\sigma_{t+1} = \sigma' \mid \sigma_t = \sigma] = \frac{\deg(\sigma \cup \sigma')}{\sum_{\sigma'' \sim \sigma} \deg(\sigma \cup \sigma'')}.$$

The last generalization required is for the auxiliary graphs G_i, $0 \leq i < d$ (Definition 1.2). For any two i-faces $\sigma, \sigma' \in X(i)$ such that $\sigma \sim \sigma'$, instead of having one edge between the corresponding vertices in G_i, we put $\deg(\sigma \cup \sigma')$ edges between them. Now the random walk on G_i is equivalent to the random walk on dimension i of X, i.e., for any $t \in \mathbb{N}$ it holds that $\pi_{t+1} = \pi_t(D_i A_i)$, where π_t is the probability distribution after t steps of the random walk on dimension i of X, and D_i, A_i are the diagonal and adjacency matrices of G_i, respectively.

1.6 Bounding the smallest eigenvalue

The last technical issue, mentioned in remark 1.1, is to make sure that $\lambda_{|V|}$ is bounded away from -1. For random walks on dimension $i \geq 1$ we can do that by another combinatorial measure of the graph, which we describe in this section.

Let $G = (V, E)$ be an undirected graph. A *bipartite component* in G is a non-empty subset of vertices $\emptyset \neq S \subseteq V$ and a partition of S into two disjoint subsets $S_1 \cup S_2 = S$ such that $|E(S, \bar{S})| = |E(S_1)| = |E(S_2)| = 0$. ($E(S_i)$ denotes the set of edges with both endpoints in S_i.) It is known that $\lambda_{|V|} = -1$ if and only if the graph has a bipartite component. Trevisan [21] has proven that when the graph is "far" from having a bipartite component, then $\lambda_{|V|}$ is bounded away from -1.

For any tuple (S, S_1, S_2), $\emptyset \neq S \subseteq V$, $S_1 \cup S_2 = S$, the *bipartiteness ratio* of (S, S_1, S_2) is defined as

$$\beta(S, S_1, S_2) = \frac{|E(S, \bar{S})| + 2(|E(S_1)| + |E(S_2)|)}{\mathrm{vol}(S)},$$

and the bipartiteness ratio of the graph is defined as

$$\beta(G) = \min_{\substack{\emptyset \neq S \subseteq V \\ S_1 \cup S_2 = S}} \beta(S, S_1, S_2).$$

It is easy to see that $\beta(G) = 0$ if and only if G has a bipartite component. Moreover, when $\beta(G)$ is close to 0, then G has a component which is "almost" bipartite. As $\beta(G)$ is bounded away from 0, G is further away from having a bipartite component. The following lemma of Trevisan [21] formalizes it.

▶ **Lemma 1.11.** *Let $G = (V, E)$ be an undirected graph where multiple edges are allowed, $\widetilde{A}$ its normalized adjacency matrix and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{|V|}$ the eigenvalues of $\widetilde{A}$. Then*

$$\lambda_{|V|} \geq -1 + \frac{\beta(G)^2}{2}.$$

When dealing with a random walk dimension 0, i.e., the vertices of the complex, we cannot guarantee that the graph is far from being bipartite. In this case we add $\deg(v)$ self-loops to any vertex $v \in V_0$. Then we get a *lazy* random walk on the vertices of the complex, where at each step we stay put with probability 1/2. It turns out that this technique cancels the negative eigenvalues of the complex. The following proposition follows from elementary linear algebra and can also be found at [18].

▶ **Proposition 1.12.** *Let $G = (V, E)$ be an undirected graph where multiple edges are allowed, $\widetilde{A}$ its normalized adjacency matrix and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{|V|}$ the eigenvalues of $\widetilde{A}$. Denote by $G' = (V, E')$ the modified graph after adding $\deg(v)$ self-loops to any vertex $v \in V$. Then the eigenvalues λ'_2 and $\lambda'_{|V|}$ of the modified normalized adjacency matrix $\widetilde{A}'$ satisfy $\lambda'_2 = (1+\lambda_2)/2$ and $\lambda'_{|V|} \geq 0$.*

1.7 Organization of this paper

The paper is organized as follows. In Section 2 we prove that the expansion of the links imply colorful expansion (Theorem 1.7). In Section 3 we prove the rapid mixing criterion for the high order random walks (Theorems 1.8 and 1.4). In Section 4 we show that Ramanujan complexes satisfy this criterion and hence form an explicit family of bounded degree high dimensional simplicial complexes in which all of the high order random walks converge rapidly to their stationary distribution (Theorem 1.9).

2 Colorful expansion

In this section we prove that the expansion of all the links of a complex imply its colorful expansion. The following is the formal version of Theorem 1.7 from the introduction.

▶ **Theorem 2.1** (Colorful expansion criterion)**.** *Let X be a d-dimensional α-skeleton expander. If $\alpha < (\sqrt[2^d]{2} - 1)/\sqrt{2}$, then there exists an $\epsilon = \epsilon(d, \alpha)$ such that X is an ϵ-colorful expander. Specifically, it holds for*

$$\epsilon = \left(\frac{\sqrt[2^d]{2} - 1 - \sqrt{2}\alpha}{2\sqrt{2}d} \right)^d.$$

This section is organized as follows. In Section 2.1 we define the required ingredients of the proof in a formal way. In Section 2.2 we just state the three propositions which are the main building blocks in the proof of the theorem. In Section 2.3 we prove the theorem assuming these propositions hold, and then in Section 2.4 we prove the main propositions which we stated earlier.

2.1 Definitions

Since we prove a global expansion property from the local expansion of the links, it is useful for us to relate cochains in the complex to their local views in the links.

▶ **Definition 2.2** (Localization)**.** Let X be a d-dimensional simplicial complex. For any i-cochain $W \in C^i(X)$, $0 \leq i \leq d$, and any j-face $\sigma \in X(j)$, $j < i$, the *localization* of W to the link of σ is defined as

$$W_\sigma = \{\tau \in X_\sigma \mid \tau \cup \sigma \in W\}.$$

Recall that for any i-cochain $W \in C^i(X)$ it holds that $\|W\| = \Pr[P_i \in W]$. When considering the localization of W to a link of some j-face $\sigma \in X(j)$ $(j < i)$ we have

$$\|W_\sigma\|_\sigma = \Pr[P_i \in W \mid P_j = \sigma].$$

This holds because the condition $P_j = \sigma$ implies that P_d is uniformly distributed on the d-faces containing σ, which are the maximal faces in the link of σ.

For any i-cochain $W \in C^i(X)$ and any dimension $j \leq i$ we define the j-cochain of fat faces. These faces are defined recursively such that any i-face in W is fat, and any j-face, $j < i$, is fat if many of the $(j+1)$-faces which contain it are fat.

▶ **Definition 2.3** (Fat faces)**.** Let X be a d-dimensional simplicial complex, $0 \leq i \leq d$, $W \in C^i(X)$ an arbitrary i-cochain and $0 < \eta < 1$ a fatness constant. The i-cochain of fat faces is defined as $S^i(W) = W$, and for any $-1 \leq j < i$

$$S^j(W) = \left\{\sigma \in X(j) \mid \left\|S^{j+1}(W)_\sigma\right\|_\sigma \geq \eta^{2^{i-j-1}}\right\}.$$

Recall that the expanding faces of any i-cochain $W \in C^i(X)$ are the $(i+1)$-faces which contain at least one face in W and one face not in W. For a simple representation of the expanding faces we define the following cochains.

▶ **Definition 2.4** (Container)**.** Let X be a d-dimensional simplicial complex. For any i-cochain $W \in C^i(X)$, $0 \leq i \leq d$, the *container* of W is defined as

$$\Gamma(W) = \{\tau \in X(i+1) \mid \exists\, \sigma \subset \tau : \sigma \in W\}.$$

▶ **Definition 2.5** (Full faces)**.** Let X be a d-dimensional simplicial complex. For any i-cochain $W \in C^i(X)$, $0 \leq i < d$, the *full faces* of dimension j, $0 \leq j \leq i+1$, are defined as

$$F^j(W) = \{\tau \in X(j) \mid \forall\, \sigma \subset \tau,\ \dim(\sigma) = j-1 : \sigma \in S^{j-1}(W)\}.$$

In words, the container of W is the $(i+1)$-cochain of faces which contain at least one i-face in W. The full faces are defined for any dimension $j \leq i+1$ and they are the j-faces which all of their $(j-1)$-faces are fat. Then we get

$$\psi(W) = \Gamma(W) \setminus F^{i+1}(W).$$

2.2 Main propositions

The first proposition ensures that, with the right parameters, there exists some dimension $j \leq i$ in which many fat j-faces contain many non-fat $(j-1)$-faces.

▶ **Proposition 2.6** (Existence of dimension in which many fat faces contain many non-fat faces)**.** *Let X be a d-dimensional simplicial complex, $0 \le i < d$, $0 < \eta < 1$ a fatness constant. Then for any i-cochain $W \in C^i(X)$, $\|W\| < \eta^{2^{i+1}-1}$, and any $c \le 1/2$ there exists $0 \le j \le i$ such that*

$$\Pr\left[P_j \in S^j(W) \wedge P_{j-1} \notin S^{j-1}(W)\right] \ge \frac{c^j}{i+1}\|W\|.$$

We relate to these non-fat $(j-1)$-faces as faces with a large "potential" since they have many fat j-faces in their links. We show that this potential can be "lifted" up to dimension $i+1$ in order to deduce that many faces of W contain these non-fat $(j-1)$-faces.

▶ **Proposition 2.7** (Large norm of fat faces implies large norm of the container of W)**.** *Let X be a d-dimensional simplicial complex, $0 \le i < d$, $0 < \eta < 1$ a fatness constant. Then for any i-cochain $W \in C^i(X)$ and any $0 \le j \le i$*

$$\Pr[P_{i+1} \in \Gamma(W) \wedge P_{j-1} \notin S^{j-1}(W)] \ge \eta^{2^{i-j}-1} \Pr[P_j \in S^j(W) \wedge P_{j-1} \notin S^{j-1}(W)].$$

The last proposition shows that, in a skeleton expander complex, the non-fat $(j-1)$-faces can be "lifted" up to dimension $i+1$ in order to deduce that many of the $(i+1)$-faces, which contain these non-fat $(j-1)$-faces, are not full.

▶ **Proposition 2.8** (Large norm of non-fat faces implies large norm of non-full faces)**.** *Let X be an α-skeleton expander, $0 \le i < d$, $0 < \eta < 1$ a fatness constant. Then for any i-cochain $W \in C^i(X)$ and any $0 \le j \le i$*

$$\begin{aligned}\Pr[P_{i+1} \in F^{i+1}(W) \wedge P_{j-1} \notin S^{j-1}(W)] \le \\ \left(\eta^{2^{i-j}} + \alpha\right) \Pr[P_j \in S^j(W) \wedge P_{j-1} \notin S^{j-1}(W)] + \\ \sum_{k=j+1}^{i} (k+1)\left(\eta^{2^{i-k}} + \alpha\right) \Pr[P_k \in S^k(W) \wedge P_{k-1} \notin S^{k-1}(W)]\end{aligned}$$

2.3 Proof of Theorem 2.1

Recall that for colorful expansion we need that any i-cochain with norm up to $1/2$ (of any dimension $i < d$) will have many expanding faces. Let $W \in C^i(X)$, $0 \le i < d$, be an arbitrary i-cochain with $\|W\| \le 1/2$. Set $\eta = \sqrt[2^{i+1}]{1/2}$ and

$$c = \frac{(\eta^{-1}-1)\eta^{2^i} - \alpha}{2(i+1)}.$$

By Proposition 2.6 there exists a dimension $0 \le j \le i$ in which many fat j-faces contain many non-fat $(j-1)$-faces, i.e.,

$$\Pr[P_j \in S^j(W) \wedge P_{j-1} \notin S^{j-1}(W)] \ge \frac{c^j}{i+1}\|W\|. \tag{1}$$

Let j be the maximal which satisfies (1), so for any $k \in \{j+1, \ldots, i\}$

$$\Pr[P_k \in S^k(W) \wedge P_{k-1} \notin S^{k-1}(W)] < \frac{c^k}{i+1}\|W\|. \tag{2}$$

We are going to show that the non-fat $(j-1)$-faces imply many expanding $(i+1)$-faces. By definition we have

$$\begin{aligned}\|\psi(W)\| = \left\|\Gamma(W) \setminus F^{i+1}(W)\right\| = \Pr[P_{i+1} \in \Gamma(W) \setminus F^{i+1}(W)] \geq \\ \Pr[P_{i+1} \in \Gamma(W) \setminus F^{i+1}(W) \wedge P_{j-1} \notin S^{j-1}(W)] \geq \\ \Pr[P_{i+1} \in \Gamma(W) \wedge P_{j-1} \notin S^{j-1}(W)] - \Pr[P_{i+1} \in F^{i+1}(W) \wedge P_{j-1} \notin S^{j-1}(W)], \quad (3)\end{aligned}$$

where both of the inequalities follow from laws of probability. The meaning of this is that the expanding faces of W are at least the difference between the container of W and the full $(i+1)$-faces, when both are restricted to the non-fat $(j-1)$-faces.

By the existence of many fat j-faces in the links of the non-fat $(j-1)$-faces we know that a large part of W is seen in these links, or in other words, we can "lift" the fat j-faces in order to deduce a lower bound on the container of W. This lower bound is given formally by Proposition 2.7:

$$\Pr[P_{i+1} \in \Gamma(W) \wedge P_{j-1} \notin S^{j-1}(W)] \geq \eta^{2^{i-j}-1} \Pr[P_j \in S^j(W) \wedge P_{j-1} \notin S^{j-1}(W)]. \quad (4)$$

By the skeleton expansion of X we can "lift" the non-fat $(j-1)$-faces in order to deduce an upper bound on the full $(i+1)$-faces. So by Proposition 2.8 we have

$$\begin{aligned}\Pr[P_{i+1} \in F^{i+1}(W) \wedge P_{j-1} \notin S^{j-1}(W)] \leq \\ (\eta^{2^{i-j}} + \alpha) \Pr[P_j \in S^j(W) \wedge P_{j-1} \notin S^{j-1}(W)] + \\ \sum_{k=j+1}^{i} (k+1)(\eta^{2^{i-k}} + \alpha) \Pr[P_k \in S^k(W) \wedge P_{k-1} \notin S^{k-1}(W)] \quad (5)\end{aligned}$$

Substituting (4) and (5) in (3) yields

$$\begin{aligned}\|\psi(W)\| \geq \left((\eta^{-1}-1)\eta^{2^{i-j}} - \alpha\right) \Pr[P_j \in S^j \wedge P_{j-1} \notin S^{j-1}(W)] - \\ \sum_{k=j+1}^{i} (k+1)(\eta^{2^{i-k}} + \alpha) \Pr[P_k \in S^k \wedge P_{k-1} \notin S^{k-1}(W)]. \quad (6)\end{aligned}$$

Now by substituting (1) and (2) in (6) we get

$$\begin{aligned}\|\psi(W)\| \geq \left(\frac{(\eta^{-1}-1)\eta^{2^{i-j}} - \alpha}{i+1} c^j - \sum_{k=j+1}^{i} \frac{k+1}{i+1} (\eta^{2^{i-k}} + \alpha) c^k \right) \|W\| \\ \geq c^j \left(\frac{(\eta^{-1}-1)\eta^{2^i} - \alpha}{i+1} - \sum_{k=1}^{i-j} \frac{k+j+1}{i+1} c^k \right) \|W\| \geq c^j (2c-c) \|W\| = c^{j+1} \|W\|,\end{aligned}$$

where the second inequality holds since

$$\eta^{2^{i-k}} + \alpha \leq \eta + \alpha < \sqrt[2^{i+1}]{1/2} + \frac{\sqrt[2^{i+1}]{2} - 1}{\sqrt{2}} = 2^{-x} + \frac{2^x - 1}{\sqrt{2}} \leq 1$$

for $x = (1/2)^{i+1} \in \left(0, 1/2\right]$, and the last inequality holds trivially for $j = i$, and for $j < i$ it holds since

$$\begin{aligned}\sum_{k=1}^{i-j} \frac{k+j+1}{i+1} c^k \leq c \left(\frac{i+1-(i-j-1)}{i+1} + \sum_{k=2}^{i-j} c^{k-1} \right) \\ \leq c \left(\frac{i+1-(i-j-1)}{i+1} + (i-j-1)\frac{1}{i+1} \right) \leq c.\end{aligned}$$

Since $j \leq i < d$ it follows that for any i-cochain $W \in C^i(X)$, $0 \leq i < d$, $\|W\| \leq 1/2$,

$$\frac{\|\psi(W)\|}{\|W\|} \geq c^{i+1} \geq \epsilon$$

for ϵ as in the theorem. ◀

2.4 Proofs of the main propositions

For ease of notation, in this section we use the following shortcuts.

- When the cochain W is clear from the context we write just S^j instead of $S^j(W)$ to denote the fat j-faces, as well as F^j and Γ instead of $F^j(W)$ and $\Gamma(W)$, respectively.
- We relate to each cochain as the event that the matching random variable is in the cochain. For instance, instead of writing $\Pr[P_j \in S^j]$ and $\Pr[P_j \notin S^j]$ we write just $\Pr[S^j]$ and $\Pr[\overline{S^j}]$, respectively.

2.4.1 Proof of Proposition 2.6

Our aim is to show that there exists a dimension $j \leq i$ in which many fat j-faces contain many non-fat $(j-1)$-faces. We first show that a large norm of fat faces of any dimension imply a large norm of W.

▶ **Lemma 2.9** (Large norm of fat faces implies large norm of W)**.** *Let X be a d-dimensional simplicial complex, $0 \leq i < d$, $0 < \eta < 1$ a fatness constant. Then for any i-cochain $W \in C^i(X)$ and any $-1 \leq j \leq i$*

$$\|W\| \geq \eta^{2^{i-j}-1} \big\|S^j\big\|.$$

Proof. Fix an i-cochain $W \in C^i(X)$ and some $-1 \leq j \leq i$. By laws of probability, for any $k \leq i$

$$\begin{aligned} \big\|S^k\big\| &= \Pr\big[S^k\big] \geq \Pr\big[S^k \wedge S^{k-1}\big] = \Pr\big[S^k \mid S^{k-1}\big] \Pr\big[S^{k-1}\big] \\ &\geq \eta^{2^{i-k}} \Pr\big[S^{k-1}\big] = \eta^{2^{i-k}} \big\|S^{k-1}\big\|, \end{aligned} \tag{7}$$

where the last inequality follows from the definition of fat faces.

Applying (7) iteratively for $k = i, i-1, \ldots, j+1$ yields

$$\|W\| \geq \eta\, \eta^2 \cdots \eta^{2^{i-j-1}} \big\|S^j\big\| = \eta^{2^{i-j}-1} \big\|S^j\big\|,$$

◀

Now we show that for any dimension $j \leq i$ the norm of W is bounded by the norm of the fat j-faces (which represent fat i-faces on fat $(i-1)$-faces on fat $(i-2)$-faces and all the way down to fat j-faces) plus the norm of fat k-faces which contain non-fat $(k-1)$-faces for all $k \in \{j+1, \ldots, i\}$.

▶ **Lemma 2.10** (Norm of W as fat faces on fat faces plus fat faces on non-fat faces)**.** *Let X be a d-dimensional simplicial complex, $0 \leq i < d$. Then for any i-cochain $W \in C^i(X)$ and any $-1 \leq j \leq i$*

$$\|W\| \leq \big\|S^j\big\| + \sum_{k=j+1}^{i} \Pr\big[S^k \wedge \overline{S^{k-1}}\big].$$

Proof. Fix an i-cochain $W \in C^i(X)$ and some $-1 \leq j \leq i$. By laws of probability, for any $k \leq i$

$$\begin{aligned}\left\|S^k\right\| &= \Pr\left[S^k\right] = \Pr\left[S^k \wedge S^{k-1}\right] + \Pr\left[S^k \wedge \overline{S^{k-1}}\right] \\ &\leq \Pr\left[S^{k-1}\right] + \Pr\left[S^k \wedge \overline{S^{k-1}}\right] = \left\|S^{k-1}\right\| + \Pr\left[S^k(W) \wedge \overline{S^{k-1}}\right]. \end{aligned} \tag{8}$$

Applying (8) iteratively for $k = i, i-1, \ldots, j+1$ finishes the proof. ◀

Now we use the above two lemmas in order to prove the proposition.

Proof of Proposition 2.6. Fix an i-cochain $W \in C^i(X)$, $\|W\| < \eta^{2^{i+1}-1}$. First we show that the empty set is not fat. By Lemma 2.9 it holds that

$$\|W\| \geq \eta^{2^{i+1}-1}\left\|S^{-1}\right\|.$$

Since $\|W\| < \eta^{2^{i+1}-1}$ it follows that $\left\|S^{-1}\right\| < 1$, which implies that $\left\|S^{-1}\right\| = 0$ because there is only one (-1)-face (the empty set). Now, if for any $0 \leq j \leq i$

$$\Pr\left[S^j \wedge \overline{S^{j-1}}\right] < \frac{c^j}{i+1}\|W\|,$$

then by Lemma 2.10

$$\|W\| \leq \sum_{j=0}^{i} \Pr\left[S^j \wedge \overline{S^{j-1}}\right] < \sum_{j=0}^{i} \frac{c^j}{i+1}\|W\| \leq \|W\|,$$

which leads to a contradiction. ◀

2.4.2 Proof of Proposition 2.7

Fix an i-cochain $W \in C^i(X)$ and some $0 \leq j \leq i$. By laws of probability it follows that

$$\Pr\left[\Gamma \wedge \overline{S^{j-1}}\right] \geq \Pr\left[\Gamma \wedge \overline{S^{j-1}} \wedge W\right] = \Pr\left[\Gamma \mid \overline{S^{j-1}} \wedge W\right] \Pr\left[\overline{S^{j-1}} \wedge W\right] = \Pr\left[\overline{S^{j-1}} \wedge W\right], \tag{9}$$

where the last equality holds since $\Pr[\Gamma \mid W] = 1$.

By an argument similar to Lemma 2.9 (just add the event $\overline{S^{j-1}}$ to each step) we get that

$$\Pr\left[W \wedge \overline{S^{j-1}}\right] \geq \eta^{2^{i-j}-1} \Pr\left[S^j \wedge \overline{S^{j-1}}\right]. \tag{10}$$

Combining (9) and (10) finishes the proof. ◀

2.4.3 Proof of Proposition 2.8

We want to show that many of the $(i+1)$-faces, which contain non-fat $(j-1)$-faces, are not full. First we count the full $(i+1)$-faces by two terms: Full $(i+1)$-faces which contain only full faces, and full $(i+1)$-faces which contain a non-full k-face for some $k \in \{j, \ldots, i\}$.

▶ **Lemma 2.11** (Full $(i+1)$-faces as full faces on full faces plus full faces on non-full faces)**.** *Let X be a d-dimensional simplicial complex, $0 \leq i < d$, $0 < \eta < 1$ a fatness constant. Then for any i-cochain $W \in C^i(X)$ and any $0 \leq j \leq i$*

$$\Pr\left[F^{i+1} \wedge \overline{S^{j-1}}\right] \leq \Pr\left[F^{j+1} \wedge \overline{S^{j-1}}\right] + \sum_{k=j+2}^{i+1} \Pr\left[F^k \wedge \overline{F^{k-1}}\right].$$

Proof. Fix an i-cochain $W \in C^i(X)$ and some $0 \leq j < i$. By laws of probability, for any $k > j$

$$\begin{aligned}\Pr\left[F^k \wedge \overline{S^{j-1}}\right] &= \Pr\left[F^k \wedge \overline{S^{j-1}} \wedge F^{k-1}\right] + \Pr\left[F^k \wedge \overline{S^{j-1}} \wedge \overline{F^{k-1}}\right] \\ &\leq \Pr\left[F^{k-1} \wedge \overline{S^{j-1}}\right] + \Pr\left[F^k \wedge \overline{F^{k-1}}\right]. \end{aligned} \tag{11}$$

Applying (11) iteratively for $k = i+1, i, \ldots, j+2$ finishes the proof. ◀

Now we want to bound the second term of Lemma 2.11, so we relate the norm of full k-faces which contain a non-full $(k-1)$-face to the norm of full k-faces which contain a non-fat $(k-2)$-face.

▶ **Lemma 2.12** (Full k-faces with a non-full $(k-1)$-face related to full k-faces with a non-fat $(k-2)$-face)**.** *Let X be a d-dimensional simplicial complex, $0 \leq i < d$, $0 < \eta < 1$ a fatness constant. Then for any i-cochain $W \in C^i(X)$ and any $1 \leq k \leq i+1$*

$$\Pr\left[F^k \wedge \overline{F^{k-1}}\right] \leq k \Pr\left[F^k \wedge \overline{S^{k-2}}\right].$$

Proof. Fix an i-cochain $W \in C^i(X)$ and some $1 \leq k \leq i+1$. Note that if a $(k-1)$-face $\sigma \in X(k-1)$ is not full, then by definition it contains a non-fat $(k-2)$-face $\tau \subset \sigma$, $\tau \in X(k-2) \setminus S^{k-2}$. Since each $(k-1)$-face contains k vertices, by removing one vertex from σ uniformly at random there is a probability of $1/k$ to hit τ. It follows that

$$\begin{aligned} 1 &= \Pr\left[\overline{F^{k-1}} \mid F^k \wedge \overline{S^{k-2}}\right] = \frac{\Pr\left[\overline{F^{k-1}} \wedge F^k \wedge \overline{S^{k-2}}\right]}{\Pr\left[F^k \wedge \overline{S^{k-2}}\right]} \\ &= \frac{\Pr\left[\overline{S^{k-2}} \mid F^k \wedge \overline{F^{k-1}}\right] \Pr\left[F^k \wedge \overline{F^{k-1}}\right]}{\Pr\left[F^k \wedge \overline{S^{k-2}}\right]} \geq \frac{\Pr\left[F^k \wedge \overline{F^{k-1}}\right]}{k \Pr\left[F^k \wedge \overline{S^{k-2}}\right]}, \end{aligned} \tag{12}$$

where the first equality holds since if $P_{k-2} \in \overline{S^{k-2}}$, then by definition P_{k-1} must have been not full, the second and third equalities follow from laws of probability and the inequality follows from the explanation above. ◀

The next lemma is the *only* place where we use the skeleton expansion of the complex. We use it in order to bound the norm of full k-faces which contain a non-fat $(k-2)$-face with relation to the norm of fat $(k-1)$-faces which contain a non-fat $(k-2)$-face.

▶ **Lemma 2.13** (Full k-faces with a non-fat $(k-2)$-face related to fat $(k-1)$-faces with a non-fat $(k-2)$-face)**.** *Let X be a d-dimensional α-skeleton expander, $0 \leq i < d$, $0 < \eta < 1$ a fatness constant. Then for any i-cochain $W \in C^i(X)$ and any $1 \leq k \leq i+1$*

$$\Pr\left[F^k \wedge \overline{S^{k-2}}\right] \leq \left(\eta^{2^{i-k+1}} + \alpha\right) \Pr\left[S^{k-1} \wedge \overline{S^{k-2}}\right].$$

Proof. Fix an i-cochain $W \in C^i(X)$ and some $1 \leq k \leq i+1$. Recall that a k-face is full if all of its $(k-1)$-faces are fat. Thus, at the link of any $(k-2)$-face, a full k-face seems as an edge between two fat vertices. It follows that for any non-fat $(k-2)$-face $\sigma \in X(k-2) \setminus S^{k-2}$

$$\left\|F^k_\sigma\right\|_\sigma \leq \left\|E(S^{k-1}_\sigma)\right\|_\sigma \leq \left(\eta^{2^{i-k+1}} + \alpha\right) \left\|S^{k-1}_\sigma\right\|_\sigma,$$

where the second inequality follows from the skeleton expansion of X and that σ is not fat. Then by the law of total probability

$$\begin{aligned}
\Pr\left[F^k \wedge \overline{S^{k-2}}\right] &= \sum_{\sigma \in X(k-2)\setminus S^{k-2}} \Pr\left[F^k \mid \sigma\right] \Pr[\sigma] \\
&= \sum_{\sigma \in X(k-2)\setminus S^{k-2}} \left\|F_\sigma^k\right\|_\sigma \Pr[\sigma] \\
&\leq \sum_{\sigma \in X(k-2)\setminus S^{k-2}} \left(\eta^{2^{i-k+1}} + \alpha\right) \left\|S_\sigma^{k-1}\right\|_\sigma \Pr[\sigma] \\
&= \left(\eta^{2^{i-k+1}} + \alpha\right) \sum_{\sigma \in X(k-2)\setminus S^{k-2}} \Pr\left[S^{k-1} \mid \sigma\right] \Pr[\sigma] \\
&= \left(\eta^{2^{i-k+1}} + \alpha\right) \Pr\left[S^{k-1} \wedge \overline{S^{k-2}}\right]. \qquad \blacktriangleleft
\end{aligned}$$

Now the proposition follows as an immediate corollary of the above lemmas.

Proof of Proposition 2.8. Fix an i-cochain $W \in C^i(X)$ and some $0 \leq j \leq i$. Then by the above lemmas

$$\begin{aligned}
\Pr\left[F^{i+1} \wedge \overline{S^{j-1}}\right] &\leq \Pr\left[F^{j+1} \wedge \overline{S^{j-1}}\right] + \sum_{k=j+2}^{i+1} \Pr\left[F^k \wedge \overline{F^{k-1}}\right] \\
&\leq \Pr\left[F^{j+1} \wedge \overline{S^{j-1}}\right] + \sum_{k=j+2}^{i+1} k \Pr\left[F^k \wedge \overline{S^{k-2}}\right] \\
&\leq \left(\eta^{2^{i-j}} + \alpha\right) \Pr\left[S^j \wedge \overline{S^{j-1}}\right] + \sum_{k=j+1}^{i} (k+1)\left(\eta^{2^{i-k}} + \alpha\right) \Pr\left[S^k \wedge \overline{S^{k-1}}\right],
\end{aligned}$$

where the inequalities follow from Lemma 2.11, Lemma 2.12 and Lemma 2.13, in that order. $\blacktriangleleft$

3 Rapid mixing of high order random walks

In this section we prove the following two theorems.

▶ **Theorem 3.1** (Colorful expansion implies rapid mixing). *Let X be a d-dimensional ϵ-colorful expander, $d > 1$. Then all of the high order random walks on X are μ-rapidly mixing for*

$$\mu = 1 - \frac{\epsilon^2}{2(d+1)^2}.$$

▶ **Theorem 3.2** (Rapid mixing criterion). *Let X be a d-dimensional α-skeleton expander, $d > 1$. If $\alpha < (\sqrt[2^d]{2} - 1)\sqrt{2}$, then all of the high order random walks on X are μ-rapidly mixing for*

$$\mu = 1 - \frac{1}{2(d+1)^2}\left(\frac{\sqrt[2^d]{2} - 1 - \sqrt{2}\alpha}{2\sqrt{2}d}\right)^{2d}.$$

We note that by following the steps of the proof carefully we actually get a stronger theorem, which we state here without proving it.

▶ **Theorem 3.3** (Stronger rapid mixing). *Let X be a d-dimensional α-skeleton expander, $d > 1$. Then for any $0 \leq i < d$ such that $\alpha < (\sqrt[2^{i+1}]{2} - 1)/\sqrt{2}$ the high order random walk on dimension i of X is μ-rapidly mixing for*

$$\mu = 1 - \frac{1}{2(i+2)^2}\left(\frac{\sqrt[2^{i+1}]{2} - 1 - \sqrt{2}\alpha}{2\sqrt{2}(i+1)}\right)^{2(i+1)}.$$

Recall that the mixing rate of a random walk is deduced from the spectrum of its normalized adjacency matrix. We show in the following lemmas that all of the induced i-graphs, $0 \leq i < d$, have a large conductance and a large bipartiteness ratio, which imply the concentration of the spectrum of the normalized adjacency matrices $\widetilde{A}_i$ for all $0 \leq i < d$.

▶ **Lemma 3.4.** *Let X be a d-dimensional ϵ-colorful expander. Then for any $0 \leq i < d$ the conductance of $G_i(X)$ satisfies*

$$\Phi(G_i(X)) \geq \frac{\epsilon}{i+2}.$$

Proof. Fix $0 \leq i < d$ and denote by $G_i = (V_i, E_i)$ the induced i-graph of X. For any i-face $\sigma \in X(i)$ and its corresponding vertex $v \in V_i$ it holds that

$$\deg(\sigma) = \frac{1}{d-i}\sum_{\substack{\tau \supset \sigma \\ |\tau| = |\sigma|+1}} \deg(\tau) = \frac{1}{(d-i)(i+1)}\sum_{\sigma' \sim \sigma} \deg(\sigma \cup \sigma') = \frac{\deg(v)}{(d-i)(i+1)}, \quad (13)$$

where the first equality holds since each d-face which contains σ contains $d-i$ $(i+1)$-faces $\tau \supset \sigma$, and hence is counted $d-i$ times. And the second equality holds since each $(i+1)$-face which contains σ contains $i+1$ neighbors of σ.

Now let $\emptyset \neq S \subseteq V_i$, $\mathrm{vol}(S)/\mathrm{vol}(V_i) \leq 1/2$, be an arbitrary subset of vertices in G_i and denote by $W \in C^i(X)$ the corresponding i-cochain in X. By (13) it follows that

$$\|W\| = \frac{\sum_{\sigma \in W}\deg(\sigma)}{\sum_{\sigma \in X(i)}\deg(\sigma)} = \frac{\sum_{v \in S}\deg(v)}{\sum_{v \in V_i}\deg(v)} = \frac{\mathrm{vol}(S)}{\mathrm{vol}(V_i)} \leq \frac{1}{2},$$

and thus by the ϵ-colorful expansion of X we get that

$$\frac{\|\psi(W)\|}{\|W\|} \geq \epsilon. \quad (14)$$

We claim that any expanding face $\tau \in \psi(W)$ contributes at least $(i+1)\deg(\tau)$ edges between S and $\bar{S}$. In order to see this, consider an expanding face $\tau \in \psi(W)$ and let $j = |\{\sigma \subset \tau \mid \sigma \in W\}|$ denote the number of i-faces of τ which are in W. Since the total number of i-faces in τ is $i+2$, then τ has $(i+2-j)$ i-faces which are not in W. It follows that there are $j(i+2-j)$ pairs of i-faces $\sigma, \sigma' \subset \tau$ such that $\sigma \in W$, $\sigma' \notin W$. Recall that for each such pair, there are $\deg(\tau)$ edges between the corresponding vertices in G_i, so τ contributes $j(i+2-j)\deg(\tau)$ edges between S and $\bar{S}$. The last thing to note is that τ is an expanding face so $1 \leq j \leq i+1$, which yields that $j(i+2-j) \geq i+1$. Therefore,

$$|E(S, \bar{S})| \geq (i+1)\sum_{\tau \in \psi(W)}\deg(\tau). \quad (15)$$

It follows that

$$\begin{aligned}\frac{\|\psi(W)\|}{\|W\|} &= \frac{\sum_{\tau \in \psi(W)}\deg(\tau)}{\sum_{\tau \in X(i+1)}\deg(\tau)} \cdot \frac{\sum_{\sigma \in X(i)}\deg(\sigma)}{\sum_{\sigma \in W}\deg(\sigma)} = \frac{i+2}{d-i} \cdot \frac{\sum_{\tau \in \psi(W)}\deg(\tau)}{\sum_{\sigma \in W}\deg(\sigma)} \\ &\leq (i+2)\frac{|E(S, \bar{S})|}{\mathrm{vol}(S)} = (i+2)\Phi(S), \end{aligned} \quad (16)$$

where the second equality holds since

$$\frac{\sum_{\sigma\in X(i)}\deg(\sigma)}{\sum_{\tau\in X(i+1)}\deg(\tau)}=\frac{\binom{d+1}{i+1}|X(d)|}{\binom{d+1}{i+2}|X(d)|}=\frac{i+2}{d-i},$$

and the inequality holds by (13) and (15).

Combining (14) and (16) finishes the proof. ◀

▶ **Lemma 3.5.** *Let X be a d-dimensional simplicial complex. Then for any $1\le i<d$ the bipartiteness ratio of $G_i(X)$ satisfies*

$$\beta(G_i(X))\ge\frac{1}{i+2}.$$

Proof. Fix $1\le i<d$ and denote by $G_i=(V_i,E_i)$ the induced i-graph of X. Let $S,S_1,S_2\subseteq V_i$ be arbitrary subsets of vertices such that $S_1\uplus S_2=S$ and denote by $W,W_1,W_2\in C^i(X)$ the corresponding i-cochains in X.

For any $(i+1)$-face $\tau\in\Gamma(W)$ denote by $j_1(\tau)=|\{\sigma\subset\tau\mid\sigma\in W_1\}|$ the number of i-faces of τ which are in W_1. Similarly denote by $j_2(\tau)$ the number of i-faces of τ which are in W_2. Since the total number of i-faces in τ is $i+2$, then τ contains $(i+2-j_1(\tau)-j_2(\tau))$ i-faces which are not in W. It follows that

$$|E(S,\bar{S})|=\sum_{\tau\in\Gamma(W)}(j_1(\tau)+j_2(\tau))(i+2-j_1(\tau)-j_2(\tau))\deg(\tau),\tag{17}$$

and for any $k=1,2$

$$2|E(S_k)|=\sum_{\tau\in\Gamma(W)}2\binom{j_k(\tau)}{2}\deg(\tau).\tag{18}$$

Combining (17) and (18) yields

$$|E(S,\bar{S})|+2(|E(S_1)|+|E(S_2)|)=\sum_{\tau\in\Gamma(W)}\Big(\big(j_1(\tau)+j_2(\tau)\big)(i+1)-2j_1(\tau)j_2(\tau)\Big)\deg(\tau).\tag{19}$$

Consider an arbitrary $(i+1)$-face $\tau\in\Gamma(W)$. If $j_1(\tau)=0$ or $j_2(\tau)=0$, then

$$\big(j_1(\tau)+j_2(\tau)\big)(i+1)-2j_1(\tau)j_2(\tau)\ge i+1.\tag{20}$$

Otherwise,

$$\big(j_1(\tau)+j_2(\tau)\big)(i+1)-2j_1(\tau)j_2(\tau)\ge\big(j_1(\tau)+j_2(\tau)\big)\left(i+1-\frac{j_1(\tau)+j_2(\tau)}{2}\right)\ge i+1,\tag{21}$$

where the first inequality holds since $j_1(\tau)j_2(\tau)\le((j_1(\tau)+j_2(\tau))/2)^2$, and the second inequality holds since $2\le j_1(\tau)+j_2(\tau)\le i+2$.

Substituting (20) and (21) in (19) yields

$$|E(S,\bar{S})|+2(|E(S_1)|+|E(S_2)|)\ge(i+1)\sum_{\tau\in\Gamma(W)}\deg(\tau).\tag{22}$$

It also holds that

$$\begin{aligned}\mathrm{vol}(S) &= \sum_{v\in S}\deg(v) = \sum_{\sigma\in W}\sum_{\sigma'\sim\sigma}\deg(\sigma\cup\sigma')\\ &= \sum_{\tau\in\Gamma(W)}\big(j_1(\tau)+j_2(\tau)\big)(i+1)\deg(\tau) \le (i+2)(i+1)\sum_{\tau\in\Gamma(W)}\deg(\tau). \qquad (23)\end{aligned}$$

Combining (22) and (23) yields

$$\beta(S,S_1,S_2) = \frac{|E(S,\bar{S})| + 2(|E(S_1)|+|E(S_2)|)}{\mathrm{vol}(S)} \ge \frac{1}{i+2}.$$

Since S, S_1, S_2 were arbitrary it follows that

$$\beta(G_i(X)) \ge \frac{1}{i+2},$$

which finishes the proof. ◀

Proof of Theorem 3.1. The proof follows from the previous lemmas. For any $0 \le i < d$ the combination of Lemma 1.10 and Lemma 3.4 yields

$$\lambda_2(\widetilde{A}_i) \le 1 - \frac{\Phi(G_i)^2}{2} \le 1 - \frac{\epsilon^2}{2(i+2)^2} \le 1 - \frac{\epsilon^2}{2(d+1)^2}. \qquad (24)$$

For $1 \le i < d$ the combination of Lemma 1.11 and Lemma 3.5 yields

$$\lambda_{|V_i|}(\widetilde{A}_i) \ge -1 + \frac{\beta(G_i)^2}{2} \ge -1 + \frac{1}{2(i+2)^2} \ge -1 + \frac{1}{2(d+1)^2} \ge -1 + \frac{\epsilon^2}{2(d+1)^2}, \qquad (25)$$

where the last inequality holds since $\epsilon < 1$.

For $i = 0$ denote by G_0' the graph obtained by adding $\deg(v)$ self-loops to any vertex $v \in V_0$. Then by Proposition 1.12

$$\lambda_2(\widetilde{A}_0') = \frac{1}{2}(1+\lambda_2(\widetilde{A}_0)) \le \frac{1}{2}\left(1+1-\frac{\Phi(G_0)^2}{2}\right) \le 1 - \frac{\epsilon^2}{4\cdot 2^2} \le 1 - \frac{\epsilon^2}{2(d+1)^2}, \qquad (26)$$

where the first and second inequalities follow from Lemma 1.10 and Lemma 3.4, respectively, and the last inequality holds since $d > 1$. By the same proposition it also holds that $\lambda_{|V_0|}(\widetilde{A}_0') \ge 0$.

By (24), (25) and (26) it follows that for any $0 \le i < d$ the high order random walk on dimension i of X is μ-rapidly mixing for

$$\mu = 1 - \frac{\epsilon^2}{2(d+1)^2}.$$ ◀

Proof of Theorem 3.2. The proof follows immediately as a corollary of Theorem 2.1 and Theorem 3.1. ◀

4 Explicit construction

Ramanujan complexes were first defined in [15] and were explicitly constructed in [14]. For details on Ramanujan complexes we refer the readers to [13].

In this section we prove the following theorem.

▶ **Theorem 4.1** (Explicit construction). *For any $d \in \mathbb{N}$ there exists a constant $q_0 = q_0(d)$ such that if X is a d-dimensional q-thick Ramanujan complex with $q > q_0$, then there exists a constant $\mu = \mu(d, q)$ such that all of the high order random walks on X are μ-rapidly mixing.*

As been proven in [4], Ramanujan complexes are excellent skeleton expanders. Though, we need a stronger claim for the mixing of their 1-dimensional skeletons than the one appears in [4]. We can get a better bound since we need a good mixing behavior only inside a subset of vertices and not between every two subsets. We start by defining a special type of complexes.

▶ **Definition 4.2** (Partite regular complex). Let X be a d-dimensional simplicial complex. X is said to be *partite regular* if there exists a partition of its vertices to disjoint subsets $V_0 \cup V_1 \cup \cdots \cup V_d = X(0)$ such that:

- For any d-dimensional face $\sigma \in X(d)$ it holds that $\sigma \in V_0 \times V_1 \times \cdots \times V_d$.
- For any $I \subset J \subset \{0, 1, \ldots, d\}$ there exists $k_I^J \in \mathbb{N}$ such that any face $\sigma \in X \cap \prod_{i \in I} V_i$ is contained in k_I^J faces from $X \cap \prod_{j \in J} V_j$.

For a partite regular complex X, denote by $X_{(i,j)} = (V_i \cup V_j, X(1) \cap (V_i \times V_j))$ the induced graph by partitions i and j. Note that by the regularity of the complex, $X_{(i,j)}$ is a bipartite biregular graph, i.e., there exists $k_i^j, k_j^i \in \mathbb{N}$, such that any vertex $v \in V_i$ is contained in k_i^j edges and any vertex $u \in V_j$ is contained in k_j^i edges. It is known that $\lambda_1(X_{(i,j)}) = (k_i^j k_j^i)^{1/2}$, where $\lambda_1(X_{(i,j)})$ is the largest eigenvalue of the graph's adjacency matrix (see [3, Lemma 3.1] for a proof). Denote by $\widetilde{\lambda}_2(X_{(i,j)}) = \lambda_2(X_{(i,j)})/(k_i^j k_j^i)^{1/2}$ the normalized second largest eigenvalue. The following mixing lemma for bipartite biregular graphs is proven in [3].

▶ **Lemma 4.3.** *[3, Corollary 3.4] Let $G = (V_1 \cup V_2, E)$ be a bipartite biregular graph, $\widetilde{\lambda}_2(G)$ its normalized second largest eigenvalue. Then for any $S \subseteq V_1, T \subseteq V_2$*

$$\frac{|E(S,T)|}{|E|} \leq \sqrt{\frac{|S||T|}{|V_1||V_2|}}\left(\sqrt{\frac{|S||T|}{|V_1||V_2|}} + \widetilde{\lambda}_2(G)\right).$$

We use this lemma in order to prove the following proposition.

▶ **Proposition 4.4** (Skeleton mixing lemma). *Let X be a partite regular d-dimensional complex, $\widetilde{\lambda}_2(X) = \max_{0 \leq i < j \leq d} \widetilde{\lambda}_2(X_{(i,j)})$. Then for any subset of vertices $S \subseteq X(0)$*

$$\|E(S)\| \leq \|S\|(\|S\| + \widetilde{\lambda}_2(X)).$$

Proof. Let $V_0 \cup V_1 \cup \cdots \cup V_d$ be the partition of $X(0)$. Note that since X is a partite regular complex, then for any $0 \leq i \leq d$ and any two vertices $u, v \in V_i$ it holds that $\deg(u) = \deg(v)$. It is achieved by the regularity property of the complex and by setting $I = \{i\}$ and $J = [d] = \{0, 1, \ldots, d\}$. This implies that for any $0 \leq i \leq d$

$$\sum_{v \in X(0)} \deg(v) = (d+1)|X(d)| = (d+1) \sum_{v \in V_i} deg(v) = (d+1)|V_i| k_{\{i\}}^{[d]}.$$

Therefore, for any subset of vertices $S_i \subseteq V_i$

$$\|S_i\| = \frac{\sum_{v \in S_i} deg(v)}{\sum_{u \in X(0)} deg(u)} = \frac{|S_i|}{(d+1)|V_i|} \tag{27}$$

In a same way, for any $0 \leq i < j \leq d$

$$\sum_{e \in X(1)} \deg(e) = \binom{d+1}{2}|X(d)| = \binom{d+1}{2} \sum_{e \in X(1) \cap (V_i \times V_j)} deg(e)$$
$$= \binom{d+1}{2}|X(1) \cap (V_i \times V_j)|k^{[d]}_{\{i,j\}}.$$

And again for any subset of edges $S_{ij} \subseteq X(1) \cap (V_i \times V_j)$

$$\|S_{ij}\| = \sum_{e \in S_{ij}} w(e) = \frac{\sum_{e \in S_{ij}} deg(e)}{\sum_{f \in X(1)} deg(u)} = \frac{2|S_{ij}|}{d(d+1)|X(1) \cap (V_i \times V_j)|}. \tag{28}$$

Now let $S \subseteq X(0)$ be an arbitrary subset of vertices in the complex. For any $0 \leq i \leq d$ denote by $S_i = S \cap V_i$ the vertices in partition i. For any $0 \leq i < j \leq d$, (27) yields

$$\sqrt{\frac{|S_i||S_j|}{|V_i||V_j|}} = \sqrt{(d+1)^2\|S_i\|\|S_j\|} = (d+1)\sqrt{\|S_i\|\|S_j\|},$$

and (28) yields

$$\frac{|E(S_i, S_j)|}{|X(1) \cap (V_i \times V_j)|} = \frac{d(d+1)}{2}\|E(S_i, S_j)\|.$$

By Lemma 4.3 it follows that for any $0 \leq i < j \leq d$

$$\|E(S_i, S_j)\| \leq \frac{2}{d}\sqrt{\|S_i\|\|S_j\|}\left((d+1)\sqrt{\|S_i\|\|S_j\|} + \widetilde{\lambda}_2(X)\right).$$

Note that since $S = S_0 \cup S_1 \cup \cdots \cup S_d$, then $\|S\| = \sum_{i=0}^{d}\|S_i\|$. Thus, the sum $\sum_{i \neq j}\|S_i\|\|S_j\|$ is maximized when all of the subsets are equal, i.e., $\|S_i\| = \|S\|/(d+1)$ for all $0 \leq i \leq d$. It follows that

$$\|E(S)\| = \sum_{i \neq j}\|E(S_i, S_j)\| \leq \sum_{i \neq j}\frac{2}{d}\sqrt{\|S_i\|\|S_j\|}\left((d+1)\sqrt{\|S_i\|\|S_j\|} + \widetilde{\lambda}_2(X)\right)$$
$$\leq \sum_{i \neq j}\frac{2}{d}\sqrt{\frac{\|S\|^2}{(d+1)^2}}\left((d+1)\sqrt{\frac{\|S\|^2}{(d+1)^2}} + \widetilde{\lambda}_2(X)\right)$$
$$= \binom{d+1}{2}\frac{2}{d(d+1)}(\|S\| + \widetilde{\lambda}_2(X)) = \|S\|(\|S\| + \widetilde{\lambda}_2(X)), \tag{29}$$

which finishes the proof. ◀

Now, as been proven in [4], all of the links of a Ramanujan complex are partite regular and the normalized second largest eigenvalue of every induced bipartite biregular graph approaches 0 as a function of the dimension and the thickness of the complex. So we state the following lemma which is proven in [4].

▶ **Lemma 4.5.** *Let X be a d-dimensional q-thick Ramanujan complex. Then there exists a constant $C = C(d)$ such that*

$$\max_{\substack{\sigma \in X \\ i \neq j}} \widetilde{\lambda}_2((X_\sigma)_{(i,j)}) \leq \frac{C}{\sqrt{q}}.$$

We are now ready to prove the theorem of this section.

Proof of Theorem 4.1. Let $d \in \mathbb{N}$ be any dimension we want. Let $C = C(d)$ be the constant from Lemma 4.5 and set

$$q_0 = \left(\frac{\sqrt{2}C}{\sqrt[2^d]{2} - 1} \right)^2.$$

Now if X is a d-dimensional q-thick Ramanujan complex with $q > q_0$, then by Proposition 4.4 and Lemma 4.5 X is an α-skeleton expander for

$$\alpha = \frac{C}{\sqrt{q}} < \frac{\sqrt[2^d]{2} - 1}{\sqrt{2}}. \tag{30}$$

Then applying Theorem 3.2 finishes the proof. ◀

Acknowledgements. We would like to thank Roei Tell for introducing us to the probabilistic view of the norm, it made the proofs much simpler.

References

1. N. Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986.
2. I. Dinur. The pcp theorem by gap amplification. *Journal of the ACM (JACM)*, 54(3):12, 2007.
3. S. Evra, K. Golubev, and A. Lubotzky. Mixing Properties and the Chromatic Number of Ramanujan Complexes. *International Mathematics Research Notices*, 2015(22):11520–11548, 2015.
4. S. Evra and T. Kaufman. Bounded degree cosystolic expanders of every dimension. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 36–48, 2016.
5. H. Garland. p-Adic Curvature and the Cohomology of Discrete Subgroups of p-Adic Groups. *Annals of Mathematics*, 97(3):375–423, 1973.
6. M. Gromov. Singularities, Expanders and Topology of Maps. Part 2: from Combinatorics to Topology Via Algebraic Isoperimetry. *Geometric And Functional Analysis*, 20(2):416–526, 2010.
7. A. Gundert and U. Wagner. On Laplacians of Random Complexes. In *Proceedings of the Twenty-eighth Annual Symposium on Computational Geometry*, pages 151–160. ACM, 2012.
8. S. Hoory, N. Linial, and A. Wigderson. Expander Graphs and their Applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.
9. T. Kaufman, D. Kazhdan, and A. Lubotzky. Ramanujan Complexes and Bounded Degree Topological Expanders. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 484–493, 2014.
10. T. Kaufman and A. Lubotzky. High dimensional expanders and property testing. In *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 501–506, 2014.
11. L. Lovász. Random walks on graphs. *Combinatorics, Paul erdos is eighty*, 2:1–46, 1993.
12. A. Lubotzky. Expander graphs in pure and applied mathematics. *Bulletin of the American Mathematical Society*, 49(1):113–162, 2012.
13. A. Lubotzky. Ramanujan complexes and high dimensional expanders. *Japanese Journal of Mathematics*, 9(2):137–169, 2014.

14 A. Lubotzky, B. Samuels, and U. Vishne. Explicit constructions of ramanujan complexes of type $\widetilde{A}_d$. *European Journal of Combinatorics*, 26(6):965–993, 2005.

15 A. Lubotzky, B. Samuels, and U. Vishne. Ramanujan complexes of type $\widetilde{A}_d$. *Israel Journal of Mathematics*, 149(1):267–299, 2005.

16 I. Oppenheim. Isoperimetric Inequalities and topological overlapping for quotients of Affine buildings. arXiv:1501.04940, 2015.

17 O. Parzanchevski and R. Rosenthal. Simplicial complexes: spectrum, homology and random walks. arXiv:1211.6775, 2012.

18 A. Sinclair and M. Jerrum. Approximate counting, uniform generation and rapidly mixing Markov chains. *Information and Computation*, 82(1):93–133, 1989.

19 M. Sipser and D. A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.

20 D. A. Spielman. *Computationally efficient error-correcting codes and holographic proofs*. PhD thesis, Massachusetts Institute of Technology, 1995.

21 L. Trevisan. Cheeger-type Inequalities for λ_n. `http://lucatrevisan.wordpress.com/2016/02/09/cheeger-type-inequalities-for-%CE%BBn/`, 2016.

A Mixing rate for general graphs

We show here that the mixing rate of random walks on a general graph can be deduced from the spectrum of its normalized adjacency matrix. The idea is similar to the proof for regular graphs, but the details require some more carefulness.

▶ **Proposition 1.1.** *Let $G = (V, E)$ be an undirected graph on n vertices, $\widetilde{A}$ its normalized adjacency matrix, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ the eigenvalues of $\widetilde{A}$ and $\lambda = \max\{|\lambda_2|, |\lambda_n|\}$. Then for any initial probability distribution $\pi_0 \in \mathbb{R}^V$ and any $t \in \mathbb{N}$*

$$\|\pi_t - \pi\|_2 \leq \sqrt{\frac{d_{max}}{d_{min}}}\lambda^t,$$

where π_t is the probability distribution after t steps of the random walk, π is the stationary distribution, $d_{max} = \max_{v\in V}\{\deg(v)\}$ and $d_{min} = \min_{v\in V}\{\deg(v)\}$.

Proof. Recall that $\widetilde{A} = D^{1/2}AD^{1/2}$, where $D = \mathrm{diag}(1/\deg(v))_{v\in V}$ and A is the adjacency matrix of G. It follows that for any initial probability distribution $\pi_0 \in \mathbb{R}^V$ and any $t \in \mathbb{N}$

$$\pi_t = \pi_0(DA)^t = \pi_0(D^{1/2}\widetilde{A}D^{-1/2})^t = \pi_0 D^{1/2}\widetilde{A}^t D^{-1/2}.$$

We show first that the claim holds for a random walk starting from any fixed vertex and then we extend it to any initial probability distribution on the vertices.

Fix a starting vertex $v \in V$ and denote by $\mathbf{1}_v$ the probability distribution which is fixed on v, i.e., $\mathbf{1}_v(v) = 1$ and for any other vertex $u \neq v$, $\mathbf{1}_v(u) = 0$. Since $\widetilde{A}$ is a symmetric matrix, its eigenvectors $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ form an orthonormal basis for $\mathbb{R}^n$, and $\widetilde{A}$ has the spectral representation

$$\widetilde{A} = \sum_{i=1}^{n} \lambda_i \mathbf{e}_i^T \mathbf{e}_i = \sum_{i=1}^{n} \lambda_i E_i,$$

where $E_i = \mathbf{e}_i^T\mathbf{e}_i$. Note that for any i, $E_i^2 = E_i$ and for any $i \neq j$, $E_iE_j = 0$. It follows that

$$\mathbf{1}_v(DA)^t = \mathbf{1}_v D^{1/2}\widetilde{A}^t D^{-1/2} = \mathbf{1}_v D^{1/2}\Big(\sum_{i=1}^{n}\lambda_i^t E_i\Big)D^{-1/2} = \sum_{i=1}^{n}\lambda_i^t \mathbf{1}_v D^{1/2}E_i D^{-1/2}. \tag{31}$$

The first eigenvalue of $\widetilde{A}$ is the trivial one, i.e., $\lambda_1 = 1$, with corresponding eigenvector

$$\mathbf{e}_1 = \left(\sqrt{\frac{\deg(u)}{\sum_{w\in V}\deg(w)}}\right)_{u\in V}.$$

Note that for any $1 \le i \le n$, $\mathbf{1}_v E_i = \mathbf{e}_i(v)\mathbf{e}_i$. Then by substitution we get

$$\lambda_1^t \mathbf{1}_v D^{1/2} E_1 D^{-1/2} = \frac{\mathbf{1}_v E_1 D^{-1/2}}{\sqrt{\deg(v)}} = \frac{\mathbf{e}_1(v)\mathbf{e}_1 D^{-1/2}}{\sqrt{\deg(v)}} = \pi. \tag{32}$$

Combining (31) and (32) yields

$$\begin{aligned}
\left\|\mathbf{1}_v(DA)^t - \pi\right\|_2 &= \left\|\sum_{i=2}^{n} \lambda_i^t \mathbf{1}_v D^{1/2} E_i D^{-1/2}\right\|_2 \le \sqrt{\frac{d_{max}}{d_{min}}}\left\|\sum_{i=2}^{n}\lambda_i^t \mathbf{1}_v E_i\right\|_2 \\
&= \sqrt{\frac{d_{max}}{d_{min}}}\left\|\sum_{i=2}^{n}\lambda_i^t \mathbf{e}_i(v)\mathbf{e}_i\right\|_2 = \sqrt{\frac{d_{max}}{d_{min}}}\sqrt{\left\langle\sum_{i=2}^{n}\lambda_i^t \mathbf{e}_i(v)\mathbf{e}_i, \sum_{j=2}^{n}\lambda_j^t \mathbf{e}_j(v)\mathbf{e}_j\right\rangle} \\
&= \sqrt{\frac{d_{max}}{d_{min}}}\sqrt{\sum_{i=2}^{n}\lambda_i^{2t}\mathbf{e}_i(v)^2} \le \sqrt{\frac{d_{max}}{d_{min}}}\lambda^t\sqrt{\sum_{i=2}^{n}\mathbf{e}_i(v)^2} \le \sqrt{\frac{d_{max}}{d_{min}}}\lambda^t,
\end{aligned}$$

where the last equality and the last inequality follow from the orthonormality of the eigenvectors.

Now let $\pi_0 \in \mathbb{R}^V$ be any initial probability distribution on the vertices. We can write π_0 as a convex combination $\pi_0 = \sum_{v\in V}\alpha_v \mathbf{1}_v$, where $\sum_{v\in V}\alpha_v = 1$. Similarly, the stationary distribution can be written as $\pi = \sum_{v\in V}\alpha_v \pi$. Then by the triangle inequality we get

$$\begin{aligned}
\|\pi_t - \pi\|_2 = \left\|\pi_0(DA)^t - \pi\right\|_2 &= \left\|\sum_{v\in V}\alpha_v\left(\mathbf{1}_v(DA)^t - \pi\right)\right\|_2 \\
&\le \sum_{v\in V}\alpha_v\left\|\mathbf{1}_v(DA)^t - \pi\right\|_2 \le \sqrt{\frac{d_{max}}{d_{min}}}\lambda^t,
\end{aligned}$$

which finishes the proof. ◀

Real Stability Testing*

Prasad Raghavendra[1], Nick Ryder[2], and Nikhil Srivastava[3]

1 **University of California, Berkeley, USA**
`raghavendra@berkeley.edu`
2 **University of California, Berkeley, USA**
`nick.ryder@berkeley.edu`
3 **University of California, Berkeley, USA**
`nikhil@math.berkeley.edu`

Abstract

We give a strongly polynomial time algorithm which determines whether or not a bivariate polynomial is real stable. As a corollary, this implies an algorithm for testing whether a given linear transformation on univariate polynomials preserves real-rootedness. The proof exploits properties of hyperbolic polynomials to reduce real stability testing to testing nonnegativity of a finite number of polynomials on an interval.

1998 ACM Subject Classification F.2.1 Numerical Algorithms and Problems

Keywords and phrases real stable polynomials, hyperbolic polynomials, real rootedness, moment matrix, sturm sequence

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.5

1 Introduction

A univariate polynomial with real coefficients is called *real-rooted* if all of its roots are real. Multivariate generalizations of this concept, known as *hyperbolic* and *real stable* polynomials, were defined in the 50's and in the 80's in the context of Partial Differential Equations and Control Theory, respectively[1], and have since made contact with several areas of mathematics. In particular, a polynomial $p \in \mathbb{R}[x_1, \ldots, x_n]$ is called *real stable* if it has no zeros with all coordinates in the open upper half of the complex plane. These polynomials have played a central role in several recent advances in theoretical computer science and combinatorics — for instance, [1, 14, 13, 7, 5]. Each of these works relies in a critical way on (1) understanding which polynomials are real stable (2) understanding which linear operators *preserve* real-rootedness and real stability. Motivated by (1) and (2), this paper studies the following two fundamental algorithmic problems:

Problem 1. Given a bivariate polynomial[2] $p \in \mathbb{R}_n[x, y]$, is p real stable?

Problem 2. Given a linear operator $T : \mathbb{R}_n[x] \to \mathbb{R}_m[x]$, does T preserve real-rootedness?

The main result of this paper is a strongly polynomial time algorithm that solves Problem 1.

* This research was supported by NSF Grants CCF-1553751, NSF CCF-1343104 and NSF CCF-1407779.
[1] See [17] for a more detailed history.
[2] We use $\mathbb{R}_n[x_1, \ldots, x_k]$ to denote the vector space of real polynomials in $x_1, \ldots, x_k$ of degree at most n in each variable.


© Prasad Raghavendra, Nick Ryder, and Nikhil Srivastava;
licensed under Creative Commons License CC-BY

▶ **Theorem 1** (Main). *Given the coefficients of a bivariate polynomial $p \in \mathbb{R}_n[x, y]$, there is a deterministic algorithm which decides whether or not p is real stable in at most $O(n^5)$ arithmetic operations, assuming exact arithmetic.*

Part of the motivation for solving Problem 1 is the following theorem of Borcea and Branden, which shows that Problem 2 can be reduced to Problem 1.

▶ **Theorem 2** (Borcea-Branden [4]). *For every linear transformation $T : \mathbb{R}_n[x] \to \mathbb{R}_m[x]$, there is a bivariate polynomial $p \in \mathbb{R}_{\max(n,m)}[x, y]$ such that T preserves real-rootedness if and only if p is real stable. Moreover, the coefficients of p can be computed from the matrix of T in linear time.*

Thus, our main theorem immediately implies a solution to Problem 2 as well.

To give the reader a feel for the objects at hand, we remark that the set of real stable polynomials in any number of variables is a nonconvex set with nonempty interior [16]. In the univariate case, the interior of the set of real-rooted polynomials simply corresponds to polynomials with distinct roots, and its boundary contains polynomials which have roots with multiplicity greater than one. With regards to Problem 2, the prototypical example of an operator which preserves real rootedness is differentiation. Recent applications such as [15] rely on finding more elaborate differential operators with this property.

We now describe the main ideas in our algorithm. It turns out that testing bivariate real stability is equivalent to testing whether a certain *two parameter* family of polynomials is real rooted. It is not clear how to check this continuum of real-rootedness statements in strongly polynomial, or even in exponential time. To circumvent this, we use a deep convexity result from the theory of hyperbolic polynomials to reduce the two parameter family to a one parameter family of degree n polynomials, whose coefficients are themselves polynomials of degree n in the parameter. We then use a characterization of real-rootedness as postive semidefiniteness of certain moment matrices to further reduce this to checking that a finite number of univariate polynomials are *nonnegative* an interval. Finally, we solve each instance of the nonnegativity problem using Sturm sequences and a bit of algebra.

The set of polynomials nonnegative on an interval forms a closed convex cone, so the last step of our algorithm may be viewed as a strongly polynomial time membership oracle for this cone. We would not be surprised if such a result is already known (at least as folklore) but we were unable to find a concrete reference in the literature, so this component of our method may be of independent interest.

We see this result as being both mathematically fundamental, as well as useful for researchers who work with stable polyomials, particularly since many of their known applications so far (e.g.[14]) put special emphasis on properties of bivariate restrictions. More speculatively, it is possible that being able to test membership in the set of real stable polynomials is a step towards being able to optimize over them.

1.1 Related Work

Problem 1 was solved in the univariate case by C. Sturm in 1835 [19], who described a now well-known method that can be turned into a strongly polynomial quadratic time algorithm given the coefficients of p [2]. We are unaware of any published work regarding algorithms for the bivariate case or for Problem 2. We remark that following the release of this paper, Thorsten Theobald has observed (informal communication) that the quantifier elimination techniques of [3] can be used to obtain *weakly* polynomial time algorithms for Problems 1 and 2.

The paper [9] studied the problem of testing whether a bivariate polynomial is *real zero* (a special case of real stability). It reduced that problem to testing PSDness of a one-parameter family of matrices which it then suggested could be solved using semidefinite programming, but without quite proving a theorem to that effect. This work is partly inspired by ideas in [9].

The paper [12] gives semidefinite programming based algorithms that can test whether certain restricted classes of *multiaffine* polynomials are real stable (in more than 2 variables).

The problem of certifying that a univariate polynomial is nonnegative is typically stated (for instance, in lecture notes) as being the solution to a semidefinite program. If one were able to work out the appropriate error to which the SDP has to be solved, this could give a weakly polynomial time algorithm for nonnegativity, which we suspect must be known as folklore. The paper [18] analyzes a semidefinite programming based algorithm in the special case when the polynomial is nondegenerate in an appropriate sense.

2 Real Stable and Hyperbolic Polynomials

We recall below the definition of a real stable polynomial in an arbitrary number of variables.

▶ **Definition 3.** A polynomial $p \in \mathbb{R}[x_1, \ldots, x_n]$ is called *real stable* if it is identically zero[3] or if $p(z_1, \ldots, z_n) \neq 0$ whenever $\mathrm{Im}(z_i) > 0$ for all $i = 1, \ldots, n$. Equivalently, p is real stable if and only if the univariate restrictions

$$t \mapsto p(te_1 + x_1, te_2 + x_2, \ldots, te_n + x_n)$$

are real rooted whenever $e_1, \ldots, e_n > 0$ and $x_1, \ldots, x_n \in \mathbb{R}$.

The equivalence between the two formulations above is an easy exercise. Note that a univariate polynomial is real stable if and only if it is real rooted. Note that we consider the zero polynomial to be real-rooted.

We will frequently use the elementary fact that a limit of real-rooted polynomials is real-rooted, which follows from Hurwitz's theorem (see, e.g. [20, Sec. 2]), or from the argument principle.

Real Stable polynomials are closely related to the following more general class of polynomials.

▶ **Definition 4.** A homogeneous polynomial $p \in \mathbb{R}[x_1, \ldots, x_n]$ is called *hyperbolic* with respect to a point $e = (e_1, \ldots, e_n) \in \mathbb{R}^n$ if $p(e) > 0$ and the univariate restrictions

$$t \mapsto p(te + x)$$

are real rooted for all $x \in \mathbb{R}^n$. The connected component of $\{x \in \mathbb{R}^n : p(x) \neq 0\}$ containing e is called the *hyperbolicity cone* of p with respect to e, and will be denoted $K(p, e)$.

Perhaps the most familiar example of a hyperbolic polynomial is the determinant of a symmetric matrix:

$$X \mapsto \det(X)$$

[3] Some works (e.g. [4]) consider only nonzero polynomials to be stable, while others [20] include the zero polynomial. We find the latter convention more convenient.

for real symmetric X, which is hyperbolic with respect to the identity matrix since the characteristic polynomial of a symmetric matrix is always real rooted. The corresponding hyperbolicity cone is the cone of positive semidefinite matrices.

The most important theorem regarding hyperbolic polynomials says that hyperbolicity cones are *always* convex, and that hyperbolicity at one point in the cone implies hyperbolicity at every other point. Thus, hyperbolic polynomials and hyperbolicity cones may be viewed as generalizing determinants and PSD cones.

▶ **Theorem 5** (Garding [6]). *If $p \in \mathbb{R}[x_1, \ldots, x_n]$ is hyperbolic with respect to $e \in \mathbb{R}^n$ then:*

1. *$K(p, e)$ is an open convex cone.*
2. *p is hyperbolic with respect to every point $y \in K(p, e)$.*

The reason hyperbolic polynomials are relevant in this work is that real stable polynomials are essentially a special case of them.

▶ **Theorem 6** (Borcea-Branden [4]). *A nonzero bivariate polynomial $p(x, y)$ of total degree at most m is real stable if and only if its homogenization*

$$p_H(x, y, z) := z^m p(x/z, y/z)$$

is hyperbolic with respect to every point in

$$\mathbb{R}^2_{>0} \times \{0\} = \{(e_1, e_2, 0) : e_1, e_2 > 0\}.$$

Thus, real stable polynomials enjoy the strong structural properties guaranteed by Theorem 5 as well, and we exploit these in our algorithm.

3 Parameter Reduction via Hyperbolicity

In this section we use the properties of hyperbolic polynomials to reduce real stability of a bivariate polynomial to testing real rootedness of a one parameter family of polynomials.

▶ **Theorem 7** (Reduction to One-Parameter Family). *A nonzero bivarite polynomial $p \in \mathbb{R}_n[x, y]$ ofis real stable if and only if following two conditions hold:*

1. *The one-parameter family of univariate polynomials $q_\gamma \in \mathbb{R}[t]$ given by,*

$$q_\gamma(t) = p(\gamma + t, t) \in \mathbb{R}[t]$$

   are real rooted for all $\gamma \in \mathbb{R}$.
2. *The univariate polynomial*

$$t \mapsto p_H(t, 1 - t, 0)$$

   is strictly positive on the interval $(0, 1)$,

Proof. (*real-stability of p* $\implies$ (1) & (2))

By Theorem 6, p_H is hyperbolic with respect to the positive orthant $\mathbb{R}^2_{>0} \times \{0\}$. Since $(1, 1, 0) \in \mathbb{R}^2_{>0} \times \{0\}$, this implies that for all $(x, y, z) \in \mathbb{R}^3$,

$$q(t) = p_H(x + t, y + t, z)$$

is real-rooted. Setting $x = \gamma$, $y = 0$ and $z = 1$ we get that $q_\gamma(t) = p_H(\gamma + t, t, 1) = p(\gamma + t, t)$ is real-rooted for all $\gamma \in \mathbb{R}$ which is condition (1). Finally, since

$$\{(t, 1 - t, 0) | t \in (0, 1)\} \subset \mathbb{R}^2_{>0} \times \{0\}$$

and p_H is hyperbolic with respect to $\mathbb{R}^2_{>0} \times \{0\}$, it follows that $p_H(t, 1-t, 0) > 0$ for all $t \in (0, 1)$.

((1) & (2) $\implies$ *real-stability of* p)

First, we claim that the polynomial p_H is hyperbolic with respect to $(1, 1, 0)$. By (2) we have $p_H(1/2, 1/2, 0) > 0$ so homogeneity implies that $p_H(1, 1, 0) > 0$. It remains to show that $q_{x,y,z}(t) = p_H(x + t, y + t, z)$ is real-rooted for all $(x, y, z) \in \mathbb{R}^3$. First, consider the case of $(x, y, z) \in \mathbb{R}^3$ with $z \neq 0$.

$$
\begin{aligned}
&\forall (x, y, z) \in \mathbb{R}^3 \text{ with } z \neq 0, p_H(x + t, y + t, z) \text{ is real-rooted} \\
&\iff \forall (x, y, z) \in \mathbb{R}^3 \text{ with } z \neq 0, p_H(\frac{x}{z} + \frac{t}{z}, \frac{y}{z} + \frac{t}{z}, 1) \text{ is real-rooted} \\
&\iff \forall (x, y, z) \in \mathbb{R}^3 \text{ with } z \neq 0, p_H(\frac{x}{z} + t, \frac{y}{z} + t, 1) \text{ is real-rooted (replacing } t/z \text{ with } t) \\
&\iff \forall (x, y) \in \mathbb{R}^2, p_H(x + t, y + t, 1) \text{ is real-rooted} \\
&\iff \forall (x, y) \in \mathbb{R}^2, p_H(x + t, t, 1) \text{ is real-rooted (replacing } t \text{ with } t - y) \\
&\iff \forall \gamma \in \mathbb{R}, p(\gamma + t, t) \text{ is real-rooted}
\end{aligned}
$$

By Hurwitz's theorem, the limit of any sequence of real-rooted polynomials is real-rooted. Therefore, if $q_{x,y,z}(t)$ is real-rooted for all $(x, y, z) \in \mathbb{R}^3$ with $z \neq 0$ then $q_{x,y,z}(t)$ is real-rooted for all $(x, y, z) \in \mathbb{R}^3$.

Given that p_H is hyperbolic with respect to $e = (\frac{1}{2}, \frac{1}{2}, 0)$, its hyperbolicity cone $K(p_H, e)$ is a convex cone containing $(1, 1, 0)$. Condition (2) implies that the connected component of $\{x | p(x) \neq 0\}$ containing $(1, 1, 0)$ contains the open line segment from $(1, 0, 0)$ to $(0, 1, 0)$. Together, this implies that the positive quadrant $\mathbb{R}^2 \times \{0\} \subseteq K(p_H, e)$. By Theorem 6, this implies that p is real-stable. ◀

Thus, our algorithmic goal is reduced to testing whether a one-parameter family is real-rooted, and whether a given univariate polynomial is positive on an interval. We solve these problems in the sequel.

4 Real-rootedness of one-parameter families

In this section we present two algorithms for testing real-rootedness of a one-parameter family of polynomials. Both algorithms reduce this problem to verifying nonnegativity of a finite number of polynomials on the real line. The first algorithm produces n polynomials of degree roughly $O(n^3)$, and has the advantage of being very simple, relying only on elementary techniques and standard algorithms such as fast matrix multiplication and the discrete Fourier transform. The second algorithm produces n polynomials of degree roughly $O(n^2)$ and runs significantly faster, but uses somewhat more specialized (but nonetheless classical) machinery from the theory of resultants.

4.1 A Simple $O(n^{3+\omega})$ Algorithm

The first algorithm is based on the observation that real-rootedness of a single polynomial is equivalent to testing positive semidefiniteness of its moment matrix, which in turn is equivalent to testing nonnegativity of the elementary symmetric polynomials of that matrix. In the more general case of a one-parameter family, the latter polynomials turn out to be polynomials of bounded degree in the parameter, and it therefore suffices to verify that these are nonnegative everywhere.

We begin by recalling the Newton Identities, which express the moments of a polynomial in terms of its coefficients.

▶ **Lemma 8** (Newton Identities). *If*

$$p(x) = \sum_{k=0}^{n} (-1)^k x^{n-k} c_k = c_0 \prod_{i=1}^{n} (x - x_i) \in \mathbb{R}[x]$$

with $c_0 \neq 0$ is a univariate polynomial with roots $x_1, \ldots, x_n$, then the moments

$$m_k := \sum_{i=1}^{n} x_i^k$$

satisfy the recurrence:

$$m_k = (-1)^{k-1} \frac{c_k}{c_0} + \sum_{i=1}^{k-1} (-1)^{k-1+i} \frac{c_{k-i}}{c_0} m_i \qquad 0 \leq k \leq n,$$

$$m_k = \sum_{k-n}^{k-1} (-1)^{k-1+i} \frac{c_{k-i}}{c_0} m_i \qquad k > n,$$

$$m_0 = n.$$

The following consequences of Lemma 8 will be relevant to analyzing our algorithm.

▶ **Corollary 9.**

1. *The moments $m_0, \ldots, m_{2n-2}$ of a degree n polynomial can be computed from its coefficients in $O(n^2)$ arithmetic operations.*
2. *Suppose $p(x) = \sum_{k=0}^{n} (-1)^k x^{n-k} c_k(\gamma)$ is a polynomial whose coefficients are polynomials $c_0(\gamma), \ldots, c_n(\gamma) \in \mathbb{R}_d[\gamma]$ in a parameter γ. Then the moments of p are given by*

$$m_k(\gamma) = r_k(\gamma)/c_0(\gamma)^k,$$

for some polynomials $r_k \in \mathbb{R}_{dk}[\gamma]$.

Proof. The first claim follows because each application of the recurrence requires at most n arithmetic operations. For the second claim, observe that each ratio $c_{k-i}(\gamma)/c_0(\gamma)$ is a rational function with a numerator of degree at most d and denominator $c_0(\gamma)$. Thus, each application of the recurrence increases the degree of the numerator by at most d and introduces an additional c_0 in the denominator. ◀

As a subroutine, we will also need the following standard result in linear algebra.

▶ **Theorem 10** (Keller-Gehrig [11]). *Given an $n \times n$ complex matrix A, there is an algorithm which computes the characteristic polynomial of A in time $O(n^\omega \log n)$.*

We now specify the algorithm and prove its correctness.

▶ **Theorem 11.** *A polynomial $p_\gamma(x) = \sum_{k=0}^{n} (-1)^k x^{n-k} c_k(\gamma)$ is real-rooted for all $\gamma \in \mathbb{R}$ if and only if the polynomials $q_0, \ldots, q_n$ output by **SimpleRR** are nonnegative on $\mathbb{R}$. Moreover, **SimpleRR** runs in time $\tilde{O}(dn^{2+\omega} + d^2 n^3)$.*

Proof. We first show correctness. Let $m_k(p)$ denote the k^{th} moment of the roots of a polynomial. By Sylvester's theorem [2, Theorem 4.58], a real polynomial

$$p_\gamma(x) = \sum_{k=0}^{n} (-1)^k x^{n-k} c_k(\gamma)$$

is real-rooted if and only if the corresponding moment matrix

$$M(\gamma)_{k,l} := m_{k+l-2}(p_\gamma)$$

is positive semidefinite. Since ν is even and c_0 has real coefficients, we have for every $\gamma \in \mathbb{R}$ that is not a root of c_0:

$$M(\gamma) \succeq 0 \iff c_0(\gamma)^\nu M(\gamma) = H(\gamma) \succeq 0.$$

Since c_0 has only finitely many roots and a limit of PSD matrices is PSD, we conclude that

$$M(\gamma) \succeq 0 \quad \forall \gamma \in \mathbb{R} \iff H(\gamma) \succeq 0 \forall \gamma \in \mathbb{R}.$$

Note that by Corollary 9 the entries of $H(\gamma)$ are polynomials of degree at most $d(\nu + 2n - 2)$ in γ.

We now recall a well-known[4] (e.g., [10]) characterization of positive semidefiniteness as a semialgebraic condition: an $n \times n$ real symmetric matrix A is PSD if and only if $e_k(A) \geq 0$ for all $k = 1, \ldots, n$, where

$$e_k(A) = \sum_{|S|=k} \det(A_{S,S})$$

is the sum of all $k \times k$ principal minors of A. Thus, p_γ is real-rooted for all $\gamma \in \mathbb{R}$ if and only if the polynomials

$$q_k(\gamma) := e_k(H(\gamma))$$

for $k = 1, \ldots, n$ are nonnegative on $\mathbb{R}$.

Since each q_k is a sum of determinants of order at most n in $H(\gamma)$ it has degree at most n in the entries of $H(\gamma)$, and we conclude that $q_1, \ldots, q_n \in \mathbb{R}_N[\gamma]$. Thus, the q_k can be recovered by interpolating them at the N^{th} roots of unity. Since the k^{th} elementary symmetric function of a matrix is the coefficient of z^{n-k} in its characteristic polynomial, this is precisely what is achieved in Step 2.

For the complexity analysis, it is clear that Step 1 takes $O(dn^2)$ time. Constructing each Hankel matrix $H(s_i)$ takes time $O(dn + n^2)$ by Corollary 9, and computing its elementary symmetric functions via the characteristic polynomial takes time $O(n^\omega \log n)$, according to Theorem 10. Thus, the total time for each iteration is $O(n^\omega \log n + dn)$, so the time for all iterations is $O(dn^{2+\omega} \log n + d^2 n^3)$. The final step requires $O(N \log N)$ time for each e_k using fast polynomial interpolation via the discrete Fourier transform, for a total of $O(dn^3 \log n)$. Thus, the total running time is $\tilde{O}(dn^{2+\omega} + d^2 n^3)$, suppressing logarithmic factors. ◀

[4] Here is a short proof: A is PSD iff $\det(zI - A)$ has only nonnegative roots. Since A is symmetric we know the roots are real. We now observe that a real-rooted polynomial has nonnegative roots if and only if its coefficients alternate in sign.

Algorithm 1: SimpleRR

Input: $(n+1)$ univariate polynomials $c_0, \ldots, c_n \in \mathbb{R}_d[\gamma]$ with $c_0 \not\equiv 0$.
Output: n univariate polynomials $q_1, \ldots, q_n \in \mathbb{R}_{3n^2d}[\gamma]$

1. Let ν be the first even integer greater than or equal to n and let $N = nd(2n-2+\nu) = O(dn^2)$. Let $s_1, \ldots, s_N \in \mathbb{C}$ be the N^{th} roots of unity.
2. For each $i = 1, \ldots N$:
   - Compute the $n \times n$ Hankel matrix $H(s_i)$ with entries

     $$H(s_i)_{k,l} := c_0(s_i)^\nu m_{k+l-2}(p_{s_i}),$$

     by applying the Newton identities (Lemma 8).
   - Compute the characteristic polynomial

     $$\det(zI - H(s_i)) = \sum_{k=0}^{n} (-1)^k z^{n-k} e_k(H(s_i))$$

     using the Keller-Gehrig algorithm (Theorem 10).
3. For each $k = 1, \ldots, n$: Use the points $e_k(H(s_1)), \ldots, e_k(H(s_N))$ to interpolate the coefficients of the polynomial

   $$q_k(\gamma) := e_k(H(\gamma)).$$

Output $q_1, \ldots, q_n$.

4.2 A Faster $O(n^4)$ Algorithm Using Subresultants

The algorithm of the previous section is based on the generic fact that a matrix is PSD if and only if its elementary symmetric polynomials are nonnegative. In this section we exploit the fact that our matrices have a special structure – namely, they are moment matrices – to find a different finite set of polynomials whose nonnegativity suffices to certify their PSDness. These polynomials are called *subdiscriminants*, and turn out to be related to another class of polynomials called *subresultants*, for which there are known fast symbolic algorithms.

Let M_p denote the $n \times n$ moment matrix corresponding to a polynomial p of degree n. Recall that $M_p = VV^T$ where V is the Vandermonde matrix formed by the roots of p. Let $(M_p)_i$ denote the leading principal $i \times i$ minor of M_p. We define subdiscriminants of a polynomial, and then show their relation to the leading principal minors of the moment matrix. For the remainder of this section it will be more convenient to use the notation

$$p(x) = \sum_{k=0}^{n} a_k x^k$$

for the coefficients of a polynomial, with roots $x_1, \ldots, x_n$ and $a_n \neq 0$.

▶ **Definition 12.** The k^{th} *subdiscriminant* of a polynomial p is defined as

$$\texttt{sDisc}_k(p) = a_n^{2k-2} \sum_{S \subset \{1,\ldots,n\}, |S|=k} \prod_{\{i,j\} \subset S} (x_i - x_j)^2$$

▶ **Lemma 13.** *The leading principal minors of the moment matrix are multiples of the*

subdiscriminants,

$$(M_p)_i = a_n^{2-2k}\mathtt{sDisc}_k(p) = \sum_{S\subset\{1,\ldots,n\},|S|=k}\prod_{\{i,j\}\subset S}(x_i-x_j)^2$$

Proof. Let

$$V_i = \begin{bmatrix} 1 & \ldots & 1 \\ x_1 & \ldots & x_n \\ \vdots & \vdots & \vdots \\ x_1^{i-1} & \ldots & x_n^{i-1} \end{bmatrix}.$$

Then $(M_p)_i = \det(V_iV_i^T)$. By Cauchy-Binet, this determinant is the sum over the determinants of all submatrices of size $i \times i$. These submatrices are exactly the Vandermonde matrices formed by subsets of the roots of size i. Then the identity follows from the formula for the determinant of a Vandermonde matrix. ◀

Equipped with this we can provide an alternative characterization of real rootedness. Define the sign of a number, denoted $\mathtt{sgn}$ to be $+1$ if it is positive, -1 if it is negative, and 0 otherwise.

▶ **Lemma 14.** *p is real-rooted if and only if the sequence $\mathtt{sgn}(\mathtt{sDisc}_1(p)),\ldots,\mathtt{sgn}(\mathtt{sDisc}_n(p))$ is first 1's and then 0's.*

Proof. Note that since $a_n \neq 0$ we have $\mathtt{sgn}(Disc_k) = \mathtt{sgn}(a_n^{2(1-k)}Disc_k) = \mathtt{sgn}((M_p)_i)$. It is clear from the definition of the subdiscriminants that if p is real-rooted with k distinct roots then $\mathtt{sDisc}_i$ is positive if $i \leq k$ and $\mathtt{sDisc}_i = 0$ if $i > k$.

Conversely, given a polynomial p with k distinct roots, then if $i > k$ we have all the minors of size i in V_i^T contain two identical rows, and hence V_i^T does not have full rank, so $V_iV_i^T$ is singular. Let $x_1, x_2, \ldots, x_j$ be the real distinct roots of p and $y_1, \bar{y_1}, \ldots, y_l, \bar{y_l}$ be the distinct complex conjugate pairs of p where $j + l = k$. Suppose the multiplicities of x_i are n_i and y_i are m_i. Then the top left $k \times k$ submatrix of M_p is

$$= \sum_i n_i \begin{bmatrix} 1 \\ x_i \\ \vdots \\ x_i^{k-1} \end{bmatrix} \begin{bmatrix} 1 & x_i & \cdots & x_i^{k-1} \end{bmatrix} + \sum_i m_i \begin{bmatrix} 1 \\ y_i \\ \vdots \\ y_i^{k-1} \end{bmatrix} \begin{bmatrix} 1 & y_i & \cdots & y_i^{k-1} \end{bmatrix} + \begin{bmatrix} 1 \\ \bar{y_i} \\ \vdots \\ \bar{y_i}^{k-1} \end{bmatrix} \begin{bmatrix} 1 & \bar{y_i} & \cdots & \bar{y_i}^{k-1} \end{bmatrix}$$

$$= \sum_i n_i \begin{bmatrix} 1 \\ x_i \\ \vdots \\ x_i^{k-1} \end{bmatrix} \begin{bmatrix} 1 & x_i & \cdots & x_i^{k-1} \end{bmatrix} + \sum_i m_i \begin{bmatrix} 1 & 1 \\ \mathrm{Re}(y_i) & \mathrm{Im}(y_i) \\ \vdots & \vdots \\ \mathrm{Re}(y_i^{k-1}) & \mathrm{Im}(y_i^{k-1}) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ \mathrm{Re}(y_i) & \mathrm{Im}(y_i) \\ \vdots & \vdots \\ \mathrm{Re}(y_i^{k-1}) & \mathrm{Im}(y_i^{k-1}) \end{bmatrix}^T$$

This shows that this submatrix is positive definite if and only if the distinct roots are all real. Note that by Sylvester's criterion this submatrix is positive definite if and only if all the leading principal minors of size $\leq k$ are positive. ◀

We now obtain a formula for the subdiscriminants of a polynomial in terms of its coefficients. The connection is provided by another family of polynomials called the *subresultants.*

▶ **Definition 15.** Let $p = \sum_{k=0}^{n} a_k x^k$ where $a_n \neq 0$. The *kth subresultant* of p, denoted $\mathtt{sRes}_k(p, p')$ is the determinant of the submatrix obtained from the first $2n-1-2k$ columns of the following $(2n-1-2k) \times (2n-1-k)$ matrix:

$$\begin{bmatrix} a_n & \cdots & \cdots & \cdots & \cdots & a_0 & 0 & 0 \\ 0 & \ddots & & & & & \ddots & 0 \\ \vdots & \ddots & a_n & \cdots & \cdots & \cdots & \cdots & a_0 \\ \vdots & & 0 & na_n & \cdots & \cdots & \cdots & a_1 \\ \vdots & \iddots & \iddots & & & & \iddots & 0 \\ 0 & \iddots & & & & \iddots & \iddots & \vdots \\ na_n & \cdots & \cdots & \cdots & a_1 & 0 & \cdots & 0 \end{bmatrix}$$

We will use two properties of subresultants. The first is a good bound on their degree as a consequence of the determinantal formula above. The second is quick algorithm to compute them. We refer the reader to [2] for a more detailed discussion of subresultants.

In this paper we will only be interested in subresultants of a polynomial with its derivative We are interested in this because of its relation to our leading principal minors:

▶ **Lemma 16** ([2] Proposition 4.27)**.** *Let* $p(x) = \sum_{k=0}^{n} a_k x^k$ *where* $a_n \neq 0$

$\mathtt{sRes}_k(p, p') = a_n \mathtt{sDisc}_{n-k}(p)$

▶ **Corollary 17.** *Since the first column of the determinant used to define the subresultant is divisible by* a_n*, we get* $\mathtt{sDisc}_k(p)$ *is a polynomial in our coefficients* $a_n, \ldots, a_0$ *of degree at most* $2n$*.*

The benefit of studying the principal minors instead of the coefficients of the characteristic polynomial for our moment matrix is that we can use an algorithm from subresultant theory to quickly calculate all the minors at once.

▶ **Theorem 18** ([2] Algorithm 8.21)**.** *There exists an algorithm which, given a polynomial* p *of degree* n *returns a list of all of its subresultants* $\mathtt{sRes}_k(p, p')$ *for* $k = 1, \ldots, n$ *in* $O(n^2)$ *time.*

▶ Remark. Many computer algebra systems (e.g., Mathematica, Macaulay2) have built-in efficient algorithms to compute subresultants.

We now combine the above facts to obtain a crisp condition for real-rootedness of a one-parameter family. Recall that by Theorem 7, we are interested in testing when a family of polynomials $p_\gamma(x)$ are real-rooted for all $\gamma \in \mathbb{R}$, where

$$p_\gamma(x) = \sum_{k=0}^{n} a_k(\gamma) x^k$$

with $c_k \in \mathbb{R}_n[\gamma]$. Let $c_m(\gamma)$ be the highest coefficient that is not identically zero. We are only interested in the case when $m \geq 2$.

▶ **Proposition 19.** *If* $p_\gamma(x) = \sum_{k=0}^{n} x^k c_k(\gamma)$ *with* $c_k \in \mathbb{R}_d[\gamma]$*, then* $\mathtt{sDisc}_k(p_\gamma)$ *is a polynomial in* γ *of degree at most* $2dn$*.*

Proof. From our previous lemma, we know that $\mathtt{sDisc}_k$ is a polynomial in the coefficients of p of degree at most $2n$. Since each of these coefficients $c_k(\gamma)$ is a polynomial in γ of degree at most d, our result follows. ◀

We now extend our characterization of real-rootedness in terms of the signs of the principal minors of a fixed polynomial to a characterization for coefficients which are polynomials in γ.

▶ **Theorem 20.** *$p_\gamma(x)$ is real-rooted for all $\gamma \in \mathbb{R}$ if and only if there exists a k such that $\mathtt{sDisc}_i(p_\gamma)$ is a nonnegative polynomial which is not identically zero for all $i \leq k$ and $\mathtt{sDisc}_i(p_\gamma)$ is identically zero for $i > k$.*

Proof. First suppose that $p_\gamma(x)$ is real rooted for all $\gamma \in \mathbb{R}$. Observe that $c_m(\gamma)$ vanishes for at most finitely many points Z_1. Moreover, the degree m discriminant of p_γ is a polynomial in γ, and is zero for at most finitely many points — call them Z_2. Thus, for $\gamma \notin Z_1 \cup Z_2$, we know that p_γ has exactly m distinct real roots, so by Lemma 14 $\mathtt{sDisc}_i(p_\gamma)$ is strictly positive for $i \leq m$ and zero for $i > m$ on this set. By continuity this implies that $\mathtt{sDisc}_i(p_\gamma)$ is nonnegative and not identically zero on $\mathbb{R}$ for $i \leq m$, and $\mathtt{sDisc}_i(p_\gamma)$ is identically zero for $i > m$, as desired.

To prove the converse, note that for $i \leq k$, $\mathtt{sDisc}_i(p_\gamma(t))$ is not identically zero, and hence there are finitely many γ away from which $\mathtt{sDisc}_i(p_\gamma)$ is positive for all $i \leq k$, and then all zero. By Lemma 14 we get that $p_\gamma(x)$ is real rooted for all these γ. Since real-rootedness is preserved by taking limits (by Hurwitz's theorem), we conclude that $p_\gamma(x)$ is real rooted for all $\gamma \in \mathbb{R}$. ◀

Combining these observations, and using the $O(n^2)$ time algorithm to compute the subdiscriminants, we arrive at the following $O(n^4)$ time algorithm for computing all the subdiscriminants.

Algorithm 2: FastRR

Input: $(n+1)$ univariate polynomials $c_0, \ldots, c_n \in \mathbb{R}_d[\gamma]$ with $c_0 \not\equiv 0$.

Output: n univariate polynomials $q_1, \ldots, q_n \in \mathbb{R}_{2dn}[\gamma]$

1. Find distinct points $\gamma_1, \ldots, \gamma_{2dn} \in$ such that $c_m(\gamma_i) \neq 0$.
2. For each γ_i use the subresultant algorithm (Theorem 18) to compute all of the $\mathtt{sRes}_k(p_{\gamma_i})$, with $k = 1, \ldots, n$.
3. Use the above values to compute $2dn$ different values $q_k(\gamma_1), \ldots, q_k(\gamma_{2dn})$ for each of the polynomials

$$q_k(\gamma) := \mathtt{sDisc}_k(p_\gamma) = c_m(\gamma)^{-1}\mathtt{sRes}_{m-k}(p_\gamma)),$$

$k = 1, \ldots, n$.

4. Use fast interpolation to compute the coefficients of $q_1, \ldots, q_n$.

Output $q_1, \ldots, q_n$.

▶ **Theorem 21.** ***FastRR*** *runs in $O(n^4)$ time.*

Proof. Since $c_n(\gamma)$ is of degree at most d we can test $2dn + d$ points to find $2dn$ points on which $c_n(\gamma)$ doesnt vanish. Each evaluation takes $O(d)$ times, so total this takes $O(d^2 n)$ time. To compute $\mathtt{sRes}_k(p_{\gamma_i})$ for each $0 \leq k \leq n-1$ and $1 \leq i \leq 2dn$ takes $O(dn^3)$ time by Theorem 18. Then to scale all the subresultants, since we have $O(dn^2)$ data points and have already computed $c_n(\gamma_i)$ takes $O(dn^2)$ time. Finally, since the degrees of the q_k are at most $2dn$, the total time to interpolate all of them is $O(dn^2 \log n)$. ◀

5 Univariate Nonnegativity Testing

In this section, we describe an algorithm to test non-negativity of a univariate polynomial over the real line.

Let $p \in \mathbb{R}[x]$ denote a univariate polynomial of degree d. The goal of the algorithm is to test if $p(x) \geq 0$ for all $x \in \mathbb{R}$. A canonical approach for the problem would be to use a Sum-of-Squares semidefinite program to express p as a sum of squares of low-degree polynomials. Unfortunately, the resulting algorithm is not a symbolic algorithm, i.e., its runtime is not strongly polynomial in the degree d, since semidefinite programming is not known to be strongly polynomial.

We will now describe a strongly polynomial time algorithm to test non-negativity of the polynomial p. Our starting point is an algorithm to count the number of real roots of a polynomial using Sturm sequences. We refer the reader to Basu et al. [2] for a detailed presentation of Sturm sequences and algorithms to compute them. For our purposes, we will need the following lemma.

▶ **Lemma 22.** *Given a univariate polynomial $p \in \mathbb{R}[x]$, the algorithm based on computing Sturm sequences uses $O(\deg(p)^2)$ arithmetic operations to determine the number of real roots of p.*

The polynomial p is positive, i.e., $p(x) > 0$ for all $x \in \mathbb{R}$, if and only if it has no real roots. Therefore, Lemma 22 yields an algorithm to test positivity using in $O(d^2)$ arithmetic operations. To test non-negativity, the only additional complication stems from the roots of the polynomial p. We begin with a simple observation.

▶ **Fact 23.** *If $p \in \mathbb{R}[x]$ is monic then $p(x) \geq 0$ for all $x \in \mathbb{R}$ if and only if p has no real roots of odd multiplicity.*

▶ **Definition 24.** A square-free decomposition of a polynomial $p \in \mathbb{R}[x]$ of degree d, is a set of polynomials $\{a_1, \ldots, a_d\} \in \mathbb{R}[x]$ such that

$$p(x) = \prod_{i=1}^{d} a_i(x)^i \ ,$$

and each a_i has no roots with multiplicity greater than one. Alternately, for each $i \in [d]$, $a_i(x)^i$ consists of all roots of p with multiplicity exactly i.

Square-free decompositions can be computed efficiently using gcd computations. Yun [21] carries out a detailed analysis of square-free decomposition algorithms. In particular, he shows that an algorithm due to Musser can be used to compute square-free decompositions at the cost of constantly many gcd computations.

Now, we are ready to describe an algorithm to test non-negativity.

Algorithm 3: Nonnegative

Input: A monic polynomial $p \in \mathbb{R}[x]$, $\deg(p) = d$

Goal: Test if $p(x) \geq 0$ for all $x \in \mathbb{R}$.

1. Using Musser's algorithm, compute the square-free decomposition of p given by,

$$p = \prod_{i \in [d]} a_i^i$$

where $a_i \in \mathbb{R}[x]$ has no roots with multiplicity greater than 1.

2. For each $i \in [\lceil \frac{d}{2} \rceil]$
   - Using Sturm sequences, test if a_{2i-1} has real roots. If a_{2i-1} has real roots p is NOT non-negative.

Runtime

Let $T_{gcd}(d)$ denote the time-complexity of computing the gcd of two univariate polynomials of degree d. The runtime of Musser's square-free decomposition algorithm is within constant factors of $T_{gcd}(d)$. Let $S_{real}(\ell)$ denote the time-complexity of determining if a degree ℓ polynomial has no real roots. Observe that

$$\sum_i \deg(a_i) \leq \deg(p) = d$$

Since $S_{real}(\ell)$ is super-linear in ℓ, we have $\sum_{i \in [d]} S_{real}(a_i) \leq S_{real}(d)$. The run-time of the algorithm is given by $O(T_{gcd}(d) + S_{real}(d))$. Using Sturm sequences, $S_{real}(d) = O(d^2)$ elementary operations on real numbers (see [2]). Using Euclid's algorithm, $T_{gcd}(d) = O(d^2)$ elemenetary operations on real numbers. This yields an algorithm for non-negativity that incurs at most $O(d^2)$ elementary operations.

6 Conclusion and Discussion

Finally, we combine the ingredients from sections 3, 4, and 5 to obtain the proof of our main theorem.

Proof of Theorem 1. Given the coefficients of p, we can compute the coefficients of the one-parameter family in (1) of Theorem 7 in time at most $O(n^3)$. By Theorem 21, **FastRR** produces the polynomials $q_1, \ldots, q_n$ in time $O(n^4)$. We check that some final segment of these polynomials are identically zero by evaluating each one at $O(n^2)$ points. These polynomials have degree $O(n^2)$, so **Nonnegative** requires time $O(n^4)$ to check nonnegativity of each remaining one, for a total running time of $O(n^5)$.

For part (2) of Theorem 7, we simply use a Sturm sequence to ensure that there are no roots in $(0, 1)$, and then evaluate the polynomial at a single point to check that the sign is positive. ◀

The algorithm in this paper offers a starting point in the area of polynomial time algorithms for real stability. In addition to the obvious possibility of improving the running time to say $O(n^4)$ or below, several natural open questions remain:

- Can the algorithm be generalized to 3 or more variables? The bottleneck to doing this is that we do not know how to check real rootedness of 2-parameter families, or equivalently, nonnegativity of bivariate polynomials.

- Is there an algorithm for testing whether a given polynomial is hyperbolic with respect to *some* direction, without giving the direction as part of the input?
- Is there an algorithm for testing stability of bivariate polynomials with *complex* coefficients?

Perhaps leaving the realm of strongly polynomial time algorithms, the major open question in this area is the following: a famous theorem of Helton and Vinnikov [8] asserts that every bivariate real stable polynomial can be written as

$$p(x, y) = \det(xA + yB + C)$$

for some positive semidefinite matrices A, B and real symmetric C. Unfortunately, their proof does not give an efficient algorithm for finding these matrices. Can the ideas in this paper, perhaps via using SDPs to find sum-of-squares representations of certain nonnegative polynomials derived from p, be used to obtain such an algorithm?

Acknowledgments. We thank Eric Hallman, Navin Goyal, Jonathan Leake, and Bernd Sturmfels for valuable discussions. We also thank Didier Henrion for valuable correspondence regarding other algorthimic approaches to these problems, and Thorsten Theobald for pointing out the reference [3].

References

1 Nima Anari and Shayan Oveis Gharan. The kadison-singer problem for strongly rayleigh measures and applications to asymmetric tsp. *arXiv preprint arXiv:1412.1143*, 2014.
2 Saugata Basu, Richard Pollack, and Marie-Francoise Roy. *Algorithms in real algebraic geometry*, volume 20033. Springer, 2005.
3 Michael Ben-Or, Dexter Kozen, and John Reif. The complexity of elementary algebra and geometry. *Journal of Computer and System Sciences*, 32(2):251–264, 1986.
4 Julius Borcea and Petter Brändén. The lee-yang and pólya-schur programs. i. linear operators preserving stability. *Inventiones mathematicae*, 177(3):541–569, 2009.
5 Julius Borcea, Petter Brändén, and Thomas Liggett. Negative dependence and the geometry of polynomials. *Journal of the American Mathematical Society*, 22(2):521–567, 2009.
6 Lars Garding. An inequality for hyperbolic polynomials. *Journal of Mathematics and Mechanics*, 8(6):957–965, 1959.
7 Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. A randomized rounding approach to the traveling salesman problem. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 550–559. IEEE, 2011.
8 J William Helton and Victor Vinnikov. Linear matrix inequality representation of sets. *Communications on pure and applied mathematics*, 60(5):654–674, 2007.
9 Didier Henrion. Detecting rigid convexity of bivariate polynomials. *Linear Algebra and its Applications*, 432(5):1218–1233, 2010.
10 Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.
11 Walter Keller-Gehrig. Fast algorithms for the characteristics polynomial. *Theoretical computer science*, 36:309–317, 1985.
12 Mario Kummer, Daniel Plaumann, and Cynthia Vinzant. Hyperbolic polynomials, interlacers, and sums of squares. *Mathematical Programming*, 153(1):223–245, 2015.
13 Adam Marcus, Daniel A Spielman, and Nikhil Srivastava. Interlacing families i: Bipartite ramanujan graphs of all degrees. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 529–537. IEEE, 2013.

14 Adam W Marcus, Daniel A Spielman, and Nikhil Srivastava. Interlacing families ii: Mixed characteristic polynomials and the kadison–singer problem. *Annals of Mathematics*, 182(1):327–350, 2015.

15 Adam W Marcus, Daniel A Spielman, and Nikhil Srivastava. Interlacing families iv: Bipartite ramanujan graphs of all sizes. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 1358–1377. IEEE, 2015.

16 Wim Nuij. A note on hyperbolic polynomials. *Mathematica Scandinavica*, 23(1):69–72, 1969.

17 Robin Pemantle. Hyperbolicity and stable polynomials in combinatorics and probability. *arXiv preprint arXiv:1210.3231*, 2012.

18 Helfried Peyrl and Pablo A Parrilo. Computing sum of squares decompositions with rational coefficients. *Theoretical Computer Science*, 409(2):269–281, 2008.

19 Charles Sturm. *Mémoire sur la résolution des équations numériques*. 1835.

20 David Wagner. Multivariate stable polynomials: theory and applications. *Bulletin of the American Mathematical Society*, 48(1):53–84, 2011.

21 David Y.Y. Yun. On square-free decomposition algorithms. In *Proceedings of the Third ACM Symposium on Symbolic and Algebraic Computation*, SYMSAC'76, pages 26–35, New York, NY, USA, 1976. ACM. doi:10.1145/800205.806320.

Very Simple and Efficient Byzantine Agreement

Silvio Micali

CSAIL, MIT, Cambridge, USA
silvio@csail.mit.edu

Abstract

We present a very simple, cryptographic, binary Byzantine-agreement protocol that, with $n \geq 3t + 1 \geq 3$ players, at most t of which are malicious, halts in expected 9 rounds.

1998 ACM Subject Classification C2.2 Network Protocols, F.0 General

Keywords and phrases Byzantine Agreement

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.6

1 Set Up

The players communicate, in rounds, in a synchronous, point-to-point network with full connectivity. Each player i has a public key PK_i, and a corresponding secret key SK_i, of a verifiable random function. (For simplicity, we may rely on a unique-signature digital signature scheme and a hash-function H modelled as a random oracle. This way, i univocally associates to each message m the random string $H(SIG_i(m))$.) There is also a random string R, independent of the n public keys. The players, their public keys, and the string R are common knowledge to all players.

2 Adversarial Model

A honest player follows all his protocol instructions. Initially all players are honest, and remain so until he made malicious (corrupted) by a polynomial-time Adversary. At the start of *any round*, the Adversary may secretly corrupt *any player* he wants, provided that he corrupts less than $n/3$ players in total. The Adversary totally controls, and perfectly coordinates, the actions of all corrupted players, who thus may arbitrarily deviate from their protocol instructions. At each round, the Adversary immediately learns all messages sent by the honest players, and then chooses the messages sent in the same round by all currently corrupted players. However, the Adversary cannot interfere (block, alter, etc.) the messages the currently honest players send to each other. In addition, since he is computationally bounded, he cannot successfully forge the digital signature of an honest player, except with negligible probability.

3 Further Information

For more details, see https://people.csail.mit.edu/silvio/Selected Scientific Papers Distributed Computation.

Also, the author and Vinod Vikuntanathan have modified the protocol so as as to work also when the Adversary corrupt any number of players less than $n/2$. Stay tuned!

© Silvio Micali;
licensed under Creative Commons License CC-BY

Low-Complexity Cryptographic Hash Functions*†

Benny Applebaum[1], Naama Haramaty-Krasne[2], Yuval Ishai[3], Eyal Kushilevitz[4], and Vinod Vaikuntanathan[5]

1 **Tel-Aviv University, Israel**
`bennyap@post.tau.ac.il`
2 **Technion, Haifa, Israel**
`haramaty@cs.technion.ac.il`
3 **Technion, Haifa, Israel, and UCLA, Los Angeles, USA**
`yuvali@cs.technion.ac.il`
4 **Technion, Haifa, Israel**
`eyalk@cs.technion.ac.il`
5 **MIT, Cambridge, USA**
`vinodv@csail.mit.edu`

Abstract

Cryptographic hash functions are efficiently computable functions that shrink a long input into a shorter output while achieving some of the useful security properties of a random function. The most common type of such hash functions is *collision resistant* hash functions (CRH), which prevent an efficient attacker from finding a pair of inputs on which the function has the same output.

Despite the ubiquitous role of hash functions in cryptography, several of the most basic questions regarding their computational and algebraic complexity remained open. In this work we settle most of these questions under new, but arguably quite conservative, cryptographic assumptions, whose study may be of independent interest. Concretely, we obtain the following results:

- **Low-complexity CRH.** Assuming the intractability of finding short codewords in natural families of linear error-correcting codes, there are CRH that shrink the input by a constant factor and have a *constant algebraic degree* over $\mathbb{Z}_2$ (as low as 3), or even *constant output locality and input locality.* Alternatively, CRH with an arbitrary polynomial shrinkage can be computed by *linear-size* circuits.
- **Win-win results.** If low-degree CRH with good shrinkage *do not* exist, this has useful consequences for learning algorithms and data structures.

* This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing, supported by the Simons Foundation and by the DIMACS/Simons Collaboration in Cryptography through NSF grant CNS-1523467.

† The first author was partially supported by the European Union's Horizon 2020 Programme (ERC-StG-2014-2020) under grant agreement no. 639813 ERC-CLC, by an ICRC grant and by the Check Point Institute for Information Security. The second author was partially supported by a Melvin R. Berlin Fellowship in the Cyber Security Research Program. The second and third authors were partially supported by ERC starting grant 259426. The second, third and fourth authors were partially supported by ISF grant 1709/14, BSF grant 2012378, and NSF-BSF grant 2015782. The third author was additionally supported by a DARPA/ARL SAFEWARE award, NSF Frontier Award 1413955, NSF grants 1228984, 1136174, 1118096, and 1065276, and DARPA through the ARL under Contract W911NF-15-C-0205. The fifth author was partially supported by NSF Grants CNS-1350619 and CNS-1414119, Alfred P. Sloan Research Fellowship, Microsoft Faculty Fellowship, the NEC Corporation, a Steven and Renee Finn Career Development Chair from MIT, and DARPA and U.S. Army Research Office under contracts W911NF-15-C-0226. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.

© Benny Applebaum, Naama Haramaty-Krasne, Yuval Ishai, Eyal Kushilevitz, and Vinod Vaikuntanathan;
licensed under Creative Commons License CC-BY

8th Innovations in Theoretical Computer Science Conference (ITCS 2017).
Editor: Christos H. Papadimitrou; Article No. 7; pp. 7:1–7:31

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- **Degree-2 hash functions.** Assuming the conjectured intractability of solving a random system of quadratic equations over $\mathbb{Z}_2$, a uniformly random degree-2 mapping is a *universal one-way hash function* (UOWHF). UOWHF relaxes CRH by forcing the attacker to find a collision with a random input picked by a challenger. On the other hand, a uniformly random degree-2 mapping is *not* a CRH. We leave the existence of degree-2 CRH open, and relate it to open questions on the existence of degree-2 randomized encodings of functions.

1998 ACM Subject Classification F.0 Theory of Computation – General

Keywords and phrases Cryptography, hash functions, complexity theory, coding theory

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.7

1 Introduction

This work studies the problem of minimizing the complexity of cryptographic hash functions. We start with some relevant background.

Cryptographic hash functions are efficiently computable functions that shrink a long input into a shorter output while achieving some of the useful security properties of a random function. The main focus of this work is on *collision resistant* hash functions (CRH), which prevent an efficient attacker from finding a pair of distinct inputs x, x' on which the function has the same output.[1] However, we will also consider *universal one-way hash function* (UOWHF) [63], which relax CRH by forcing the attacker to find a collision with a random input x picked by a challenger.

CRH are among the most useful and well studied cryptographic primitives. They are commonly used in cryptographic protocols, with applications ranging from sublinear-communication and statistically hiding commitments [29, 47], via succinct and efficiently verifiable arguments for NP [54, 62], to protocols that bypass black-box simulation barriers [7]. More directly, they can be used via the "hash and sign" paradigm to reduce the task of digitally signing a long message x to the easier task of signing a short hash $h(x)$ [28, 61]. Analogously, they can reduce the cost of verifying the correctness of a long NP-statement x to that of verifying the correctness of a short NP-statement $y = h(x)$ by having the prover argue that she knows some x' such that $h(x') = y$ and x' is a true statement. Thus, the amortized cost of signing a long message or verifying a long NP-statement is essentially the cost of computing a CRH.

While the *feasibility* of CRH can be based on a variety of standard cryptographic assumptions, including the conjectured intractability of factoring, discrete logarithms, and lattice problems [28, 42, 65, 58], questions about the *efficiency* of CRH are still quite far from being settled. In particular, recent progress on the efficiency of other "symmetric" cryptographic primitives, such as pseudorandom generators, [20, 75], pseudorandom functions [43], and even UOWHFs, does not seem relevant in light of the known black-box separation between CRH and these primitives [69, 45]. The goal of the present work is to close some of the remaining gaps in our understanding of the complexity of CRH and related primitives.

[1] Technically speaking, a CRH is defined by a collection of input-shrinking functions h_z, where z is a public evaluation key, and where the security requirement should hold with respect to a randomly chosen z. This has the advantage of allowing security against non-uniform attackers. In the following presentation we will treat a CRH as a single deterministic function for simplicity.

We study the following natural complexity measures:

- **Degree.** We say that $h : \{0,1\}^k \to \{0,1\}^m$ has algebraic degree d if each output can be written as a multivariate polynomial over $\mathbb{Z}_2$ in the inputs of degree at most d. Ideally, we would like the degree to be constant, where 2 is the best one could hope for.
- **Locality.** We say that h has *output locality* d if each output depends on at most d inputs. Ideally, we would like the output locality to be constant, where output locality 3 is the best one could hope for [41]. If h has output locality d then its degree is at most d. Similarly, h has *input locality* d if every input influences at most d outputs.
- **Circuit size.** We say that h has circuit size S if it can be computed by a boolean circuit of size S over the standard AND/OR/NOT basis (with AND/OR gates of fan-in 2).[2] Ideally, we would like the circuit size to be *linear* in the input length. Linear size is implied by constant output locality.

The goals of minimizing circuit size and locality can be directly motivated by the goals of reducing the sequential and parallel time complexity of hashing. Minimizing algebraic degree, other than being of theoretical interest, is motivated by applications in which hashing is computed in the encrypted or secret-shared domain. Indeed, it is typically the case that techniques for secure multiparty computation [12, 26, 66], homomorphic encryption [39, 22, 40], or homomorphic secret sharing [27, 21] are much more efficient when applied to low-degree computations over a small field. See [46] for further discussion.

The prior state of the art can be summarized as follows. Standard algebraic or number theoretic constructions of CRH, as well as (asymptotic versions of) the commonly used practical designs, do not achieve constant degree or locality, and their circuit size is quasi-linear or worse. General techniques for randomized encoding of functions can be used to convert any standard CRH h in NC^1 into a CRH $\hat{h}$ with constant output locality [1]. However, even if h has very good shrinkage, $\hat{h}$ only shrinks the input by a sublinear amount, which limits its usefulness. From here on, we will restrict the attention by default to hash functions that have linear (or better) shrinkage, namely $h : \{0,1\}^k \to \{0,1\}^{ck}$ for some $0 < c < 1$. Every such h with linear circuit size can be converted into a linear-size CRH with polynomial shrinkage, namely $h' : \{0,1\}^k \to \{0,1\}^{k^\epsilon}$ for an arbitrary $\epsilon > 0$, using a tree of invocations of h [60]. Linear-size UOWHFs were constructed in [52] under strong assumptions. The assumptions were later improved in [3], who also achieved constant locality. The question of obtaining similar results for CRH was left open by both works. Finally, heuristic constructions of CRH with constant algebraic degree have been proposed in the literature [30]. However, the security of these proposals has not been reduced to a well studied problem.

To summarize, prior to our work, linear-shrinkage CRH candidates with constant algebraic degree were only proposed as heuristics, and no candidate CRH construction with constant locality or linear circuit size has been proposed.

1.1 Our Contribution

In this work we settle most of the open questions concerning the complexity of CRH and related primitives under new, but arguably clean and conservative, cryptographic assumptions.

Concretely, we put forward the following class of *binary SVP* assumptions. For a distribution $\mathcal{M}$ over $m \times n$ binary matrices and a parameter $0 < \delta < 1/2$, the $(\mathcal{M}, \delta) - \mathsf{bSVP}$ assumption asserts that given a matrix M drawn from $\mathcal{M}$, no efficient algorithm can find

[2] One could alternatively consider *running time* on a RAM machine; our upper bounds on circuit size apply to this model as well.

a nonzero vector in the kernel of M whose Hamming weight is at most δn, except with negligible success probability. The matrix M can be thought of as the parity-check matrix of a binary linear error-correcting code. Thus, bSVP can be viewed as a binary field analogue of the lattice Shortest Vector Problem (SVP), replacing an integer lattice by a binary code.

We construct low-complexity CRH based on instances of the bSVP assumption with matrix distributions $\mathcal{M}$ that correspond to uniform distributions over natural classes of linear codes. The parameter δ is chosen such that there are exponentially many codewords whose relative weight is close to δ, but where such codewords are only an exponentially small fraction of the set of all codewords, thus ruling out "guessing attacks." When $m = \alpha n$ and $\mathcal{M}$ is sufficiently rich (in particular, when it is uniform over *all* $m \times n$ matrices), the assumption is plausible whenever $\delta < \alpha/2$. The assumption does *not* hold when $\delta > \alpha/2$, since in this case a codeword of weight δn can be found by solving a system of linear equations.

Despite being a simple and natural cryptographic assumption, we are not aware of any explicit study or even precise formulation of the bSVP assumption in the literature. While we were not able to reduce useful instances of bSVP to any standard cryptographic assumption, we do show that such instances have a "win-win" flavor in the sense that if they are broken, this would necessarily have useful algorithmic consequences. Natural instances of the bSVP assumption are likely to find additional applications in cryptography, and their further study may be of independent interest from both a cryptography and coding theory points of view.

We now give a more detailed account of our results.

Low-complexity CRH

Assuming bSVP for a random linear code with $\delta > 2H_2^{-1}(\alpha)$ (where H_2 denotes the binary entropy function), there are CRH that shrink the input by a constant factor and have a *constant algebraic degree*. We give a direct construction of degree-5 CRH, and then reduce the degree to 3 by using a new optimized randomized encoding construction for constant-degree functions (previous randomized encoding methods from [1] can also reduce the degree to 3, but at the expense of compromising the linear shrinkage feature).

Assuming bSVP for a random low-density parity-check code (LDPC), we can also get *constant output and input locality*, which imply CRH with an arbitrary polynomial shrinkage that can be computed by *linear-size* circuits. The assumption that bSVP holds for LDPCs may look too strong in light of the fact that LDPCs admit efficient decoding algorithms. However, known decoding techniques seem to have only limited relevance to bSVP. Indeed, the known reductions from bSVP to unique decoding introduce exponential overhead (cf. [32]). Moreover, there is a gap between the noise level p for which LDPC's admit efficient decoding and the relative distance Δ of LDPC's which essentially corresponds to our parameter δ. This gap grows with the (constant) locality parameter [23], and the LDPC becomes similar to random linear code both combinatorially [37, 57], and, presumably, in terms of its intractability.[3]

Our constructions take the following natural high level approach. First the input is deterministically encoded into a longer vector that has a low weight. This encoding is done via a simple function Expand that has constant input and output locality. Then the encoded

[3] We further mention that the problem of finding (many) w-weight codewords in LDPC with sub-constant rate (e.g., when the parity check matrix has m rows and $n = O(m^{7/5})$ columns and $w = O(m^{0.2})$) was implicitly considered by Feige, Kim and Ofek [34]. In particular, it was shown that if the problem is easy (for randomly chosen 3-sparse ,matrices) then one can efficiently *refute* random 3-CNF's with m variables and $m^{1.4}$, beating the state-of-the-art refutation algorithms.

input is shrunk by applying a random linear mapping M sampled from $\mathcal{M}$, where M is used as a key specifying the CRH. Finding a collision implies finding a low-weight vector in the kernel of M (namely, the sum of two images of Expand), which is intractable if the appropriate instance of the bSVP assumption holds. A practically-oriented hash function candidate with a similar structure was proposed by Augot et al. [4] (see also [35]). In fact, our degree-5 construction can be obtained as an instance of their construction (with a specific choice of parameters). Our other instantiations of this approach are different and are tailored to different optimization goals.

As an application, our linear-size CRH imply (together with other cryptographic assumptions, cf. [16]) the first succinct non-interactive argument system for NP in which the verifier's algorithm can be implemented by a linear-size circuit in the statement length. They also imply the first linear-size implementations of non-interactive *statistically hiding commitments* (SHC), a randomized variant of CRH that can be used to hide the input. This follows from the known constructions of SHC from CRH [29, 47].

Win-win results

To gain more insight on the instances of the bSVP assumption on which we rely, we show that refuting them would have useful algorithmic consequences. Concretely, we show two types of such results. First, we show that either (1) there is a linearly-shrinking CRH with logarithmic degree (a non-trivial object that does not seem to follow from standard assumptions) or (2) one can achieve an arbitrary polynomial speedup over the celebrated BKW algorithm for Learning Parities with Noise (LPN) [19]. The latter would be considered a breakthrough in light of the large body of work on algorithms for LPN and its variants. Second, we show that breaking useful instances of bSVP, on which a degree-3 linearly-shrinking CRH can be based, leads to a surprisingly good *data structure* for learning parities from random (noiseless) examples in a natural distributed learning model.

Degree-2 hash functions

Finally, we study the case of hash functions that have the minimal possible degree. We first address the case of UOWHFs, showing that a random shrinking degree-2 mapping is a UOWHF assuming that it is one-way. The latter is equivalent to a fairly well studied assumption, known as the "MQ assumption" [59, 73], which asserts that solving a random system of quadratic equations is intractable.

We then show several results on the existence of a degree-2 CRH. We show that a *random* degree-2 shrinking function is not collision resistant, strengthening a claim from [30] that was restricted to the case of linear-shrinkage. This result can be extended to the case of SHC, leaving open the possibility of constructing degree-2 CRH and SHC by using other distributions over degree-2 mappings.

We relate this question to questions on the existence of degree-2 randomized encodings of functions that were left open by [51]. The high level idea is that while for strong version of randomized encoding the existence of degree-2 encodings for general functions can be ruled out, there are relaxed versions for which this question is still open, yet these relaxed versions are strong enough to respect the security properties of CRH and SHC. Thus, ruling out a degree-2 implementation of these primitives would require settling the above open questions in the negative.

Organization

Following some preliminaries (Section 2), in Section 3 we discuss the assumptions on which we rely, including the bSVP assumption we introduce and the MQ assumption. In Section 4 we present constructions of low-complexity CRH from variants of bSVP. In Section 5 we present our positive and negative results for degree-2 hash functions. Finally, in Section 6 we present the "win-win" results showing that if low-complexity CRH do not exist, this has useful algorithmic consequences.

2 Preliminaries

General

We let $[n]$ denote the set $\{1, \ldots, n\}$. We naturally view n-bit strings as (column) vectors over the binary field $\mathbb{Z}_2$. For a pair of strings $x, x' \in \{0,1\}^n$, we let $\Delta(x, x')$ denote the relative Hamming distance between x and x', i.e., $|\{i \in [n] : x_i \neq x'_i\}|/n$. We let $\Delta(x)$ denote the (relative) Hamming weight of x, i.e., $\Delta(x) = \Delta(x, 0^n)$. By default, logarithms are taken to base 2. For real $p \in [0,1]$ we let $H_2(p) := -p\log(p) - (1-p)\log(1-p)$ denote the binary entropy function where $0 \log 0$ is taken to be 0. The inverse of the binary entropy function, $H_2^{-1} : [0,1] \to [0, \frac{1}{2}]$, maps $y \in [0,1]$ to the unique $x \in [0, \frac{1}{2}]$ for which $H_2(x) = y$. It is well known (cf. [44, Chapter 3]) that for every constant $\delta \in (0, 1/2)$

$$2^{nH_2(\delta)-o(n)} \leq \binom{n}{\delta n} \quad \text{and} \quad \sum_{i=1}^{\delta n} \binom{n}{i} \leq 2^{nH_2(\delta)}. \tag{2.1}$$

We also use the following approximation taken from [24, Theorem 2.2]:

$$\frac{x}{2\log(6/x)} \leq H_2^{-1}(x) \leq \frac{x}{\log(1/x)}. \tag{2.2}$$

A function $\epsilon(\cdot)$ is said to be negligible if $\epsilon(k) < k^{-c}$ for any constant $c > 0$ and sufficiently large k. We will sometimes use $\mathrm{neg}(\cdot)$ to denote an unspecified negligible function. The statistical distance between two probability distributions X and Y, denoted $\mathsf{SD}(X;Y)$, is defined as the maximum, over all functions A, of the distinguishing advantage $|\Pr[A(X) = 1] - \Pr[A(Y) = 1]|$. A pair of distribution ensembles $X = \{X_k\}$ and $Y = \{Y_k\}$ is *statistically indistinguishable* if $\mathsf{SD}(X_k; Y_k) \leq \mathrm{neg}(k)$.

Locality and Degree

Let $f : \{0,1\}^k \to \{0,1\}^m$ be a function. We say that the i-th output variable y_i *depends* on the j-th input variable x_j (or equivalently, x_j *affects* the output y_i) if there exists a pair of input strings which differ only on the j-th location whose images differ on the i-th location. The locality of an output variable (resp., input variable) is the number of input variables on which it depends (resp., the number of output variables which it affects). We say that an output has degree d if it can be expressed as a multivariate polynomial of degree d in the inputs over the binary field $\mathbb{Z}_2$. The locality of an output variable trivially upper bounds its degree. The output locality (resp., degree) of f is the maximum output locality (resp., degree) over all outputs of f. Similarly, the input locality of f is the maximal input locality over all inputs of f.

▶ **Definition 1** (Collision-Resistant Hash Functions)**.** A collection of functions

$$\mathcal{H} = \left\{ h_z : \{0,1\}^k \to \{0,1\}^{m(k)} \right\}_{z \in \{0,1\}^{s(k)}}$$

is a *collision-resistance hash* (CRH) function if the following hold:

- (Shrinkage) The output length is smaller than the input length: $m(k) < k$ for every k.
- (Efficient evaluation and sampling) There exists a pair of efficient algorithms: (a) an *evaluation algorithm* H which given $(z \in \{0,1\}^s, x \in \{0,1\}^k)$ outputs $h_z(x)$; and (b) a key-sampling algorithm $\mathcal{K}$ which given 1^k samples an index $z \in \{0,1\}^{s(k)}$.
- (Collision resistance) For every probabilistic polynomial-time adversary Adv it holds that

$$\Pr_{z \xleftarrow{R} \mathcal{K}(1^k)} [\mathsf{Adv}(z) = (x, x') \text{ s.t. } x' \neq x \text{ and } h_z(x) = h_z(x')] \tag{2.3}$$

  is negligible in k.

The *shrinking factor* of $\mathcal{H}$ is the ratio m/k. We say that $\mathcal{H}$ is *linearly shrinking* if m/k is upper-bounded by a constant $c < 1$. Similarly, $\mathcal{H}$ is *polynomially shrinking* if $m/k < 1/k^c$ for some constant $c \in (0,1)$.

The weaker variant of *universal one-way hash function* (UOWHF) [63] is defined by relaxing the third item (collision resistance) with the following requirement (also known as *target collision resistance* [11]):

- (Target collision resistance) For every pair of probabilistic polynomial-time adversaries $\mathsf{Adv} = (\mathsf{Adv}_1, \mathsf{Adv}_2)$ it holds that

$$\Pr_{\substack{(x,r) \xleftarrow{R} \mathsf{Adv}_1(1^k) \\ z \xleftarrow{R} \mathcal{K}(1^k)}} [\mathsf{Adv}_2(z, x, r) = x' \text{ s.t. } x' \neq x \text{ and } h_z(x) = h_z(x')] \leq \mathrm{neg}(k).$$

That is, first the adversary Adv_1 specifies a target string x and a state information r, then a random hash function h_z is selected, and then Adv_2 tries to form a collision x' with x under h_z.

▶ Remark (Measuring efficiency). When saying that a collection $\mathcal{H}$ of hash functions enjoys some level of efficiency we refer to the complexity of *every fixed* function h_z in the collection $\mathcal{H}$. For example, $\mathcal{H}$ has constant output locality of d if every function $h_z \in \mathcal{H}$ has output locality of at most d. A stronger form of efficiency guarantees that given the index z and the input x the function $H(z,x) = h_z(x)$ has the required level of efficiency (e.g., constant locality). Since the index z is selected once and for all we adopt the former (weaker) variant as our default notion. However, some of our constructions also guarantee the stronger form of efficiency.

▶ Remark (Public coins). Our constructions are all in the "public-coin" setting [49], and so they remain secure even if the adversary gets the coins used to sample the index of the collection.

2.1 Randomized Encoding of Functions

Roughly speaking, a *randomized encoding* [51, 1] of a function $f(x)$ is a randomized mapping $\hat{f}(x;r)$ such that for every input x the output distribution $\hat{f}(x;r)$ (induced by a random choice of r) depends only on the output of $f(x)$. Throughout the paper we employ *perfect randomized encoding* as defined below.

▶ **Definition 2** (Perfect Randomized Encoding). Let $f : \{0,1\}^n \to \{0,1\}^m$ be a function. We say that a function $\hat{f} : \{0,1\}^n \times \{0,1\}^\rho \to \{0,1\}^s$ is a *perfect randomized encoding* (PRE) of f if there exists a deterministic decoding algorithm C and a randomized simulator S which satisfy the following:

- (Perfect correctness) For every input $x \in \{0,1\}^n$ and $r \in \{0,1\}^\rho$, it holds that $C(\hat{f}(x;r)) = f(x)$.
- (Perfect privacy) For every $x \in \{0,1\}^n$, the distribution $\hat{f}(x;r)$, induced by a uniform choice of $r \xleftarrow{R} \{0,1\}^\rho$, is identical to the distribution $S(f(x))$.
- (Balanced simulation) The distribution $S(y)$ induced by choosing $y \xleftarrow{R} \{0,1\}^m$ is identical to the uniform distribution over $\{0,1\}^s$.
- (Length preserving) The difference between the output length and the total input length of the encoding $s - (n + \rho)$ is equal to the difference $m - n$ between the output length and the input length of f.

We refer to the second input of $\hat{f}$ as its *random input* and to ρ and s as the *randomness complexity* and *output complexity* of $\hat{f}$, respectively.

The definition naturally extends to collections of functions $\mathcal{F} = \left\{f_z : \{0,1\}^{n(z)} \to \{0,1\}^{m(z)}\right\}_{z\in\{0,1\}^*}$. In particular, we say that $\hat{\mathcal{F}} = \left\{\hat{f}_z : \{0,1\}^{n(z)} \times \{0,1\}^{\rho(z)} \to \{0,1\}^{s(z)}\right\}_{z\in\{0,1\}^*}$ perfectly encodes $\mathcal{F}$ if for every z, $\hat{f}_z$ perfectly encodes f_z. Furthermore, we always assume that the encoding is uniform in the sense that there exists a polynomial-time algorithm which given z outputs a description (say as a boolean circuit) of the encoding $\hat{f}_z$, its decoder C_z and its simulator S_z.

In [1] it is shown that a PRE of a CRH is also CRH.

▶ **Lemma 3** ([1, Lemma 7.2]). *If $\mathcal{H} = \left\{h_z : \{0,1\}^k \to \{0,1\}^m\right\}$ is a CRH then its perfect encoding $\hat{\mathcal{H}} = \left\{\hat{h}_z : \{0,1\}^k \times \{0,1\}^\rho \to \{0,1\}^s\right\}$ is also a CRH, where $\hat{\mathcal{H}}$ is viewed as a collection of single-input functions (of input length $k + \rho$) which uses the key-sampling algorithm of $\mathcal{H}$.*

Observe that if the collection $\mathcal{H}$ has linear-shrinkage and the perfect encoding $\hat{\mathcal{H}}$ has randomness complexity of $O(k)$, then the collection $\hat{\mathcal{H}}$ also has linear shrinkage.

The following proposition follows from [1, Section 4].

▶ **Proposition 4.** Let $f : \{0,1\}^n \to \{0,1\}^\ell$ be a function that each of its outputs can be computed by a formula of size at most t. Then, f can be encoded by a PRE $\hat{f}$ with degree 3, output locality 4, randomness/output complexity of $\ell \cdot \mathrm{poly}(t)$ and input locality of at most $c \cdot \mathrm{poly}(t)$ where c is the input locality of f. In particular, if f has constant output locality then the randomness complexity of $\hat{f}$ is $O(\ell)$, and if, in addition, f has constant input locality, then $\hat{f}$ also has constant input locality.

We will also need two standard closure properties of PREs. First, just like in the case of string-encodings, if we take an encoding $\hat{f}$ of f, and re-encode it by $\hat{\hat{f}}$, then the resulting encoding also encodes the original function f. Second, given an encoding $\hat{f}(y;r)$ of $f(y)$ we can encode a function of the form $f(g(x))$ by encoding the outer function and substituting y with $g(x)$, i.e., $\hat{f}(g(x);r)$. We summarize these properties via the following lemmas (taken from [1, Lemma 4.11] and [2, Fact 3.1]).

▶ **Lemma 5** (Composition lemma). *Suppose that $g(x;r_g)$ is a PRE of $f(x)$ and $h((x,r_g);r_h)$ is a PRE of $g((x,r_g))$ (viewed as a single-argument function). Then, the function $\hat{f}(x;(r_g,r_h)) \triangleq h((x,r_g);r_h)$ is a PRE of f.*

▶ **Lemma 6** (Substitution lemma). *Suppose that the function $\hat{f}(x;r)$ is a PRE of $f(x)$. Let $h(z)$ be a function of the form $f(g(z))$ where $z \in \{0,1\}^k$ and $g : \{0,1\}^k \to \{0,1\}^n$. Then, the function $\hat{h}(z;r) \triangleq \hat{f}(g(z);r)$ is a PRE of h.*

3 Our Assumptions

3.1 The Binary Shortest Vector Problem

In the following the term *matrix sampler* refers to an algorithm $\mathcal{M}$ which, given 1^n, samples $m(n) \times n$ binary matrix for some integer-valued function $m(n)$.

▶ **Definition 7** (binary SVP)**.** For a weight parameter $\delta(n) : \mathbb{N} \to (0, 1/2)$, and an efficient sampler $\mathcal{M}(1^n)$ which samples $m(n) \times n$ binary matrices, the $(\mathcal{M}, \delta)$-bSVP assumption asserts that for every efficient algorithm Adv the probability

$$\Pr_{\mathbf{M} \stackrel{R}{\leftarrow} \mathcal{M}(1^n)} [\mathsf{Adv}(\mathbf{M}) = \mathbf{x} \text{ such that } \mathbf{x} \neq \mathbf{0}, \mathbf{Mx} = \mathbf{0} \text{ and } \Delta(\mathbf{x}) \leq \delta]$$

is negligible in n. If $m(n)/n \leq \alpha(n)$, then we refer to the distribution sampled by $\mathcal{M}$ as (α, δ)-bSVP *hard distribution*.

Using a coding-theoretic terminology, we can think of $\mathcal{M}$ as specifying an ensemble of binary linear codes which are represented by their $m \times n$ parity-check matrices. We will always assume that, except with negligible probability, the rows of $M \stackrel{R}{\leftarrow} \mathcal{M}(1^n)$ are linearly independent and so the code has rate of $1 - m/n$. The binary SVP assumption asserts that it is hard to find a short codeword (of weight at most δ) in a random member of this ensemble. For the purpose of constructing hash functions, we will choose $\delta(n)$ such that a code sampled from $\mathcal{M}(1^n)$ is likely to contain (exponentially) many codewords of weight at most $\delta(n)$ (but such light codewords capture only exponentially-small fraction of all codewords). Intuitively, this setting corresponds to the list-decoding regime of the code.[4] We mention that in the worst case, it is NP-hard to compute the distance of a linear code [72] or even to approximate it by a constant factor [33]. Currently, the best known algorithms run in exponential time (cf. [71, 31, 8, 36, 14, 10]). Let us present the main distributions used in the paper.

3.1.1 The Random Linear Code Ensemble

The ensemble of *random linear codes* is probably the most natural choice for bSVP. Formally, for a length parameter $\alpha(n) : \mathbb{N} \to (0, 1)$ and weight parameter $\delta(n)$, we let (α, δ)-bSVP denote the $(\mathcal{M}, \delta)$-bSVP assumption where $\mathcal{M}(1^n)$ uniformly samples a matrix from $\mathbb{Z}_2^{\lceil \alpha(n) \cdot n \rceil \times n}$. It is well known that a random linear code of rate $R = (1 - \alpha)$ achieves the Gilbert—Varshamov bound (cf. [44]). Specifically, for any constants δ, α for which $\delta < H_2^{-1}(\alpha)$, a uniformly chosen matrix $\mathbf{M} \stackrel{R}{\leftarrow} \mathbb{Z}_2^{\alpha n \times n}$ has no codewords of weight less than δ (except with exponentially small probability $\exp(-\Omega(n))$). Accordingly, we will be interested in the regime where $\delta > H_2^{-1}(\alpha)$. For this regime we make the following simple observations. For simplicity, we restrict our attention to the case where α and δ are constants which do not depend on n.

▶ **Observation 8** (Attack based on linear algebra)**.** *If $\delta > \alpha/2$ the (α, δ)-bSVP assumption does not hold. In particular, there is a polynomial-time algorithm that solves the problem with probability $\frac{1}{2}$.*

Proof. Given $\mathbf{M}$, we can always find a set S of $m(n) = \lceil \alpha n \rceil$ linearly-independent columns which span the column space (since the matrix has only $m(n)$ rows). We can therefore find a solution $\mathbf{x}$ of weight αn to the original system by letting $\mathbf{x}_S$ be the unique vector in the kernel of the restricted matrix $\mathbf{M}_S$, and by letting $\mathbf{x}_{[n] \setminus S}$ be the all zero vector.

[4] In fact, we show that, over random linear codes, a variant of the list-decoding problem reduces to bSVP (see Lemma 29).

To do better, we choose S so that the random variable M_S (induced by M) is a random full rank square matrix (e.g., by choosing the lexicographically-first S). In this case, the (unique) vector $\mathbf{x}_S$ in $\ker(\mathbf{M}_S)$ is uniformly distributed over $\mathbb{Z}_2^{\alpha n}$ and so with probability $\frac{1}{2}$ its relative weight is at most $\frac{1}{2}$. It follows that, with probability $\frac{1}{2}$, the algorithm outputs a vector in the kernel of $\mathbf{M}$ whose Hamming weight is at most $\alpha/2$. ◀

We do not know of any polynomial-time attack for the case $\delta < \alpha/2$.

▶ **Observation 9.** *If (α, δ)-bSVP holds for $\delta > H_2^{-1}(\alpha)$ then one-way functions exist.*

Proof. Let $m(n) = \lceil \alpha n \rceil$. Consider the algorithm S which samples a random $\mathbf{x} \in \mathbb{Z}_2^n$ of weight $\lfloor \delta n \rfloor$ and then samples a random matrix $\mathbf{M} \in \mathbb{Z}_2^{m(n) \times n}$ subject to $\mathbf{M} \cdot \mathbf{x} = \mathbf{0}$. We claim that the mapping f that takes the random coins of the algorithm and outputs $\mathbf{M}$ is one-way. Indeed, assume, towards a contradiction, that the mapping can be inverted efficiently by an adversary Adv with probability ϵ, then, we can break (α, δ)-bSVP by applying Adv on $\mathbf{M} \xleftarrow{R} \mathbb{Z}_2^{m(n) \times n}$, get r, and apply the sampler in the forward direction to recover a solution $\mathbf{x}$.

To analyze the success probability it suffices to show that the distribution sampled by S (on which Adv is promised to succeed) is statistically close to the uniform distribution over $\mathbb{Z}_2^{m(n) \times n}$. Indeed, this follows by noting that (1) S samples the uniform distribution over all matrices whose kernel contains a vector of relative weight δ; and (2) The probability that a uniformly chosen parity-check matrix $\mathbf{M} \xleftarrow{R} \mathbb{Z}_2^{m(n) \times n}$ will not have a vector of weight δ in its kernel is exponentially small (cf. [9]). This completes the proof. ◀

We will later show (Theorem 14) that hardness for $\delta > 2H_2^{-1}(\alpha)$ implies the existence of collision-resistance hash functions. Due to the above attack, this means that the weight parameter δ should be in the interval

$$(2H_2^{-1}(\alpha), \alpha/2).$$

Plugging in the approximation from Eq. (2.2), and letting $\alpha = 2^{-k}$, we conclude that δ should live in the interval

$$(2^{-k+1}/k, 2^{-k-1}),$$

so the ratio between the upper-bound and the lower-bound grows when $\alpha = 2^{-k}$ decreases.

3.1.2 Random LDPC Ensemble

Instead of taking a uniformly-chosen parity-check matrix, one can use an ensemble of Low-Density Parity-Check Codes (LDPC) [37]. Concretely, for constant $\alpha \in (0, 1)$ and constant $d \in \mathbb{N}$ for which $c = \alpha d$ is an integer, we let (d, α, δ)-bSVP denote the $(\mathcal{M}_{\alpha,d}, \delta) - \mathsf{bSVP}$ assumption where $\mathcal{M}_{\alpha,d}(1^n)$ samples a uniformly chosen $\alpha \cdot n \times n$ matrix subject to the constraint that each column contains exactly c ones and each row contains exactly d ones.[5]

This ensemble of codes is well studied in the coding theory literature. Most notably, it is known that, for any fixed α and $d > 2$, a typical code in the ensemble can be efficiently decoded in the presence of constant noise rate of $p(\alpha, d) > 0$ (say over the binary symmetric channel) [37]. So in some regime of parameters, the unique decoding problem over LDPCs

[5] We implicitly restrict our attention to n's for which $(c/d) \cdot n$ is an integer, and, correspondingly, assume hardness only for these input lengths. Nevertheless, since this set of inputs is sufficiently dense, we can derive collision resistant hash functions for all input lengths via standard padding.

can be solved efficiently. Interestingly, for a fixed α, larger sparsity d reduces the noise rate $p(\alpha, d)$ for which known efficient decoding techniques work [23], whereas, combinatorially, when d grows the code becomes better and approaches the performance of a random linear code [37, 57].

Several methods for finding codewords of minimal weight in LDPC's have been proposed (see [48, 50, 74, 53, 32] and references therein). It is typically unknown how to analyze the complexity of these heuristics but an experimental study seems to suggest that the complexity grows exponentially with the distance of the code which is linear in the block length n (see also the discussion in [32]).

Overall, one may conjecture that when sparsity grows the intractability of binary SVP over LDPC codes "approaches" the intractability of SVP over the random linear code ensemble. Specifically, it seems plausible that for some constant α and every $\delta < \alpha/2$ there exists a (sufficiently large) constant d for which (d, α, δ)-bSVP holds.

3.2 Multivariate Quadratic Assumptions

We first define the multivariate quadratic (MQ) assumption that we use in this work.

▶ **Definition 10.** Let $\mathcal{D} = \{\mathcal{D}_{n(\lambda),m(\lambda),p(\lambda)}\}_{\lambda\in\mathbb{N}}$ be an ensemble of probability distributions that output a sequence of $m = m(\lambda)$ *upper triangular* matrices $\mathbf{Q}_1, \ldots, \mathbf{Q}_m \in \mathbb{Z}_p^{n\times n}$ (where we will write p for $p(\lambda)$ and n for $n(\lambda)$ from now on), and m vectors $\mathbf{L}_1, \ldots, \mathbf{L}_m \in \mathbb{Z}_p^n$. The $\mathcal{D}$-multivariate quadratic assumption (which we will refer to simply as the MQ assumption, when the parameter $\mathcal{D}$ is obvious from the context) states that it is computationally hard to find a *non-zero* solution to a given set of m quadratic equations

$$\left\{ q_i(\mathbf{x}) \stackrel{\Delta}{=} \mathbf{x}^T\mathbf{Q}_i\mathbf{x} + \mathbf{L}_i^T\mathbf{x} \stackrel{\Delta}{=} \sum_{j,k\in[n]} q_{i,j,k}x_jx_k + \sum_{j\in[n]} \ell_{i,j}x_j = 0 \pmod p \right\}_{i\in[m]}$$

where $q_{i,j,k}$ are the (j,k)-th entries of the matrix $\mathbf{Q}_i$ and $\ell_{i,j}$ are the j-th entries of the vector $\mathbf{L}_i$.

3.2.1 Previous Work on the MQ Problem

It is well-known that the multivariate quadratic (MQ) problem is NP-hard in the worst case [38]. To the best of our knowledge, the best algorithms for the *random* MQ problem with $m = O(n)$ run in $2^{\Omega(n)}$ time. Kipnis, Patarin and Goubin [55] showed that a random instance of MQ can be solved in polynomial time if $m(n) = O(\sqrt{n})$. Kipnis and Shamir [56] showed that the MQ problem can be solved in polynomial time when $m(n) = \Omega(n^2)$. The hardness of the MQ function for a randomly chosen instance is not known to follow from any well-studied intractability assumption.

Regarding cryptographic usefulness, it is easy to see that the average-case hardness of the MQ problem immediately gives us a one-way function. In the regime where $m > n$, the same assumption also gives us a pseudorandom generator [13].

The early work using the MQ assumption, starting from Matsumoto and Imai [59], focused on the (much) harder task of constructing public-key encryption schemes. The hardness of the MQ problem was necessary but not sufficient for the semantic security of their encryption scheme. Indeed, their proposal was attacked [64, 56] and fixed many times. However, none of the attacks break the MQ assumption, but rather were the result of the additional structure introduced into the assumption to obtain public-key cryptography. We do not go into the details of this long line of work as public-key cryptography is not the focus of our paper.

However, a reader interested in the history of this early work is referred to Christopher Wolf's Ph.D. thesis [73].

Moving on to hashing, the topic of this paper, Aumasson and Meier [5] showed that a *random* and *sparse* degree-2 function is not collision-resistant. Ding and Yang [30] claim that the sparsity condition can be removed, namely that a random degree-2 function with compression level $m(n) = n/2$ is not collision-resistant; however, no formal proof of this claim was provided. (Jumping ahead, we will show that indeed, their intuition was correct and that one can find in polynomial time collisions in the random MQ function with *any* non-trivial compression.) Ding and Yang [30] also conjectured that a random degree-3 function with the same compression level is a CRH. Billet, Robshaw and Peyrin [15] observed that given the difference between a possible colliding inputs of degree-2 function, the collision can be found in polynomial time. This method was first presented by Patarin in [64].

We are not aware of any results on universal one-way hashing from MQ-type assumptions.

4 Hash Functions from the Binary SVP Assumption

4.1 A General Template

We show how to construct CRH based on the bSVP assumption. Our CRH is keyed with a matrix $\mathbf{M} \in \mathbb{Z}_2^{m \times n}$, it first takes an input $\mathbf{x} \in \{0,1\}^k$ and expand it into an n-bit vector $\mathbf{y}$ via some preprocessing mapping Expand, and then compresses $\mathbf{y}$ to an m-bit vector $\mathbf{z}$ by computing $\mathbf{My}$. Formally, the construction has the following structure.

▶ **Construction 11.** Let $n = n(k)$ and $m = m(k)$ be integer valued functions. For a mapping $\mathsf{Expand} : \{0,1\}^k \to \{0,1\}^n$, and a matrix-sampler $\mathcal{M}(1^n)$ which samples $m \times n$ matrices, we define the collection of functions $\mathcal{H}_{k,m} = \{h_{\mathbf{M}} : \{0,1\}^k \to \{0,1\}^m \ : \ \mathbf{M} \in \mathbb{Z}_2^{m \times n}\}$ where

$$h_{\mathbf{M}}(\mathbf{x}) = \mathbf{M} \cdot \mathsf{Expand}(\mathbf{x}),$$

and the key-sampling algorithm $\mathcal{K}(1^k)$ outputs $\mathbf{M} \xleftarrow{R} \mathcal{M}(1^n)$.

We say that a function $\mathsf{Expand} : \{0,1\}^k \to \{0,1\}^n$ is (b,β)-*expanding* if (1) the function is injective; (2) the expansion factor n/k is at most b; and (3) the function outputs strings whose relative Hamming weight is at most β.

▶ **Lemma 12.** *Suppose that Construction 11 is instantiated with a matrix sampler $\mathcal{M}(1^n)$ which samples an (α,δ)-*bSVP *hard distribution and with a (b,β)-expanding algorithm* Expand *where $b\alpha < 1$ and $2\beta \leq \delta$. Then, the resulting collection $\mathcal{H}$ is a collision-resistance hash function with shrinkage factor of $b\alpha$.*

Proof. First observe that since $b\alpha < 1$ the function $h_{\mathbf{M}}$ is shrinking, as required. Next, we show that a collision finder Adv can be used to find a short vector in the kernel of $\mathbf{M}$. Assume, towards a contradiction, that there exists an efficient collision-finder Adv that given an $m(k) \times n(k)$ matrix $\mathbf{M} \xleftarrow{R} \mathcal{K}(1^k)$ outputs a collision $\mathbf{x} \neq \mathbf{x}' \in \mathbb{Z}_2^k$ with non-negligible probability $\epsilon(k)$. We show that the vector $\mathbf{y} = \mathsf{Expand}(\mathbf{x}) \oplus \mathsf{Expand}(\mathbf{x}')$ is (1) non-zero vector (2) it has relative weight of at most δ, and (3) it is in $\ker(\mathbf{M})$. Indeed, (1) follows since Expand is injective, (2) follows since the image of Expand contains only strings whose relative Hamming is at most β and so the vector $\mathbf{y}$ has Hamming weight of at most $2\beta \leq \delta$. Finally, since the pair $(\mathbf{x},\mathbf{x}')$ forms a collision, it holds that $\mathbf{M} \cdot \mathsf{Expand}(\mathbf{x}) = \mathbf{M} \cdot \mathsf{Expand}(\mathbf{x}')$ and therefore (3) follows as well. ◀

In the following we show that one can construct (b, β) expanding algorithm with constant input and output locality (and therefore also with constant degree and linear circuit size). The first part of the lemma optimizes the parameters b and β (and achieves large locality parameters) and the second part optimizes locality (at the expense of a looser relation between b and β).

▶ **Lemma 13.** *For every constant $\beta \in (0, 1/2)$ (weight upper-bound) the following holds.*

1. *For every $b > 1/H_2(\beta)$ there exists an efficiently computable mapping $\mathsf{Expand} : \{0,1\}^k \to \{0,1\}^{n(k)}$ which is (b, β)-expanding for all sufficiently large k's and has constant input locality c and constant output locality d where c and d depend on β and b.*
2. *If β is a power of $1/2$ then there exists an efficiently computable $(b = \frac{1}{\beta \log(1/\beta)}, \beta)$-expanding mapping $\mathsf{Expand}' : \{0,1\}^k \to \{0,1\}^{n(k)}$ with output locality of $\log(1/\beta)$ and input locality of $1/\beta$.*

Note that $b > 1/H_2(\beta)$ is a necessary requirement (otherwise by Eq. (2.1) the image of the mapping contains less than 2^k strings and so it cannot be injective). Therefore, the first part of the lemma achieves an optimal dependency between b and β.

Proof. (1) Without the locality constraint, the algorithm $\mathsf{Expand}_{\beta,b}$ can be easily implemented. Indeed, given an input $x \in \{0,1\}^k$ interpreted as an integer in $[1, 2^k]$, we can output the lexicographically x-th n-bit string of weight w efficiently by computing the output $y \in \{0,1\}^n$ in a bit-by-bit manner. (E.g., compute the number $T = \binom{n-1}{w}$ of n-bit word of weight w that begin with zero, set y_1 to zero if $x < T$ and to 1 otherwise, and continue recursively with the other bits.) Setting $n = \lfloor kb \rfloor$ and $w = \lfloor \beta n \rfloor$, and recalling that, by Eq. (2.1), for $b > 1/H_2(\beta)$ and sufficiently large k, there are more than 2^k strings of length n and weight w, we conclude that the mapping is injective.

To get low locality choose sufficiently large constants c and d such that $1/H_2(\beta) < c/d < b$ and such that $\mathsf{Expand}_{\beta,c/d}$ is injective for inputs of length d. Now partition your k-bit input into k/d-blocks of size d each. Apply $\mathsf{Expand}_{\beta,c/d}$ to each such block of inputs and generate an output block of length at most c. Output the concatenation of all k/d output blocks. By definition, the mapping has the desired expansion and locality. The (b, β)-expansion property is inherited from the original $\mathsf{Expand}_{\beta,c/d}$ algorithm.

(2) The procedure Expand' works as follows: It splits the k input bits into blocks of size $\log(1/\beta)$ bits. For each block $\mathbf{z} \in \{0,1\}^{\log(1/\beta)}$, compute a block $\mathbf{z}' \in \{0,1\}^{1/\beta}$ which is 1 in exactly the $\mathbf{z}^{th}$ location. It is easy to check that the output length n is $\frac{k}{\beta \log(1/\beta)}$ and that its Hamming weight is $\frac{k}{\log(1/\beta)} = \beta n$, as required. It is also clearly injective and can be computed with output locality of $\log(1/\beta)$ and input locality of $1/\beta$. ◀

4.2 Degree-3 CRH

Based on Lemmas 12 and 13, we prove the following theorem.

▶ **Theorem 14.** *Suppose that there exist constants $\delta \in (0, 1/4), \alpha \in (0, 1)$ with $\delta > 2H_2^{-1}(\alpha)$ and an efficient matrix sampler $\mathcal{M}$ which samples some (α, δ)-bSVP hard distribution. Then, there exists a linearly shrinking CRH with constant degree.*

Proof. Fix δ and α and take $\beta = \delta/2$ and $b \in (1/H_2(\beta), 1/\alpha)$ (the interval is non-empty due to the requirement $\delta > 2H_2^{-1}(\alpha)$). Instantiate Construction 11 with the matrix sampler $\mathcal{M}$ and the (b, β)-expanding mapping Expand promised in item 1 of Lemma 13. Then, by Lemma 12, we derive a linearly-shrinking CRH with constant degree. ◀

Our next goal is to turn the above construction into a degree-3 CRH with linear shrinkage. To this end, we show that the CRH constructed in Theorem 14 admits a degree-3 perfect randomized encoding that uses only linear amount ($O(k)$) of random bits.

▶ **Lemma 15.** *Let $h(\mathbf{x})$ be a function of the form $\mathbf{M} \cdot g(\mathbf{x})$ where $g : \mathbb{Z}_2^k \to \mathbb{Z}_2^n$ is an NC^0 function and $\mathbf{M}$ is an $m \times n$ matrix of full rank with $m < n$. Then, h can be perfectly encoded by a degree-3 function with randomness complexity of $O(n)$.*

Recall that any function can be perfectly encoded by a degree-3 encoding (Proposition 4), however the randomness complexity of the best known general transformations grows polynomially with the total size of the formulas (or branching programs) that compute the output bits of the encoded function. Such a blowup will give us only sub-linear shrinkage. The lemma bypasses this problem by introducing a new randomness-efficient degree-3 encoding which is tailored to the structure of the CRH constructed in Theorem 14.

Proof. Let $\ell = n - m$ and take $\mathbf{L} \in \mathbb{Z}_2^{n \times \ell}$ to be a matrix which spans the kernel of $\mathbf{M}$. We begin by observing that the function h is perfectly encoded by the function

$$\hat{h}(\mathbf{x}; \mathbf{r}) = g(\mathbf{x}) + \mathbf{L}\mathbf{r},$$

where $\mathbf{r} \in \mathbb{Z}_2^\ell$. Indeed, given $\mathbf{z} = \hat{h}(\mathbf{x}; \mathbf{r})$ we can decode $h(\mathbf{x})$ by computing $\mathbf{M} \cdot \mathbf{z}$. On the other direction, given $\mathbf{y} = h(\mathbf{x})$ we can perfectly simulate $\mathbf{z} = \hat{h}(\mathbf{x}; \mathbf{r})$ by sampling a random preimage of $\mathbf{y}$ under $\mathbf{M}$. Since $\mathbf{M}$ has full rank, the resulting simulator is balanced. Finally, the encoding is stretch preserving since the output-input difference $n - (k + \ell)$ of $\hat{h}$ equals to $m - k$, the output-input difference of h.

Next consider the NC^0 function $f(\mathbf{x}, \mathbf{y})$ which takes $\mathbf{x} \in \mathbb{Z}_2^k$ and $\mathbf{y} \in \mathbb{Z}_2^n$ and outputs $g(\mathbf{x}) + \mathbf{y}$. By Proposition 4, f can be perfectly encoded by a degree-3 encoding $\hat{f}(\mathbf{x}, \mathbf{y}; r')$ with randomness complexity of $O(ns^2) = O(n)$.

Finally, observe that the function $\hat{h}(\mathbf{x}; \mathbf{r})$ can be written as $f(\mathbf{x}, \mathbf{L}\mathbf{r})$. Therefore, by the substitution lemma (Lemma 6), the function $e(\mathbf{x}, \mathbf{r}; r') \triangleq \hat{f}(\mathbf{x}, \mathbf{L}\mathbf{r}; r')$ perfectly encodes the function $\hat{h}(\mathbf{x}, \mathbf{r})$. Hence, by the composition lemma (Lemma 5), the function $e(\mathbf{x}; \mathbf{r}, r')$ perfectly encodes h. Noting that e has degree 3 and randomness complexity of $O(n+\ell) = O(n)$, the lemma follows. ◀

Recall that we always assume that a hard-bSVP-distribution puts all but a negligible fraction of its mass on matrices whose columns are linearly independent. Hence, we can apply Lemma 15 to the CRH constructed in Theorem 14, and derive, by Lemma 3, the following improved version of Theorem 14.

▶ **Theorem 16.** *Under the hypothesis of Theorem 14, there exists a linearly shrinking CRH with degree 3.*

Instantiating bSVP with the random linear code ensemble, we derive the following corollary.

▶ **Corollary 17.** *Suppose that there exist constants $\delta \in (0, 1/4), \alpha \in (0, 1)$ which satisfy $\delta > 2H_2^{-1}(\alpha)$ for which $(\alpha, \delta) - \mathsf{bSVP}$ holds. Then, there exists a degree-3 linearly-shrinking CRH.*

Corollary 17 serves as a feasibility result for the existence of degree-3 CRH with linear shrinkage. This statement attempts to optimize the parameters of the underlying intractability assumption (making it as plausible as possible) and the degree, but yields a poor (constant) shrinkage factor. By iterating the CRH a sufficiently large (constant) number of times, we

can reduce the shrinkage factor to an arbitrary constant ϵ at the expense of increasing the degree to a large constant $d = d(\epsilon)$. Alternatively, we can get a better tradeoff between the degree and the shrinkage factor by strengthening the assumption as follows.

▶ **Proposition 18.** Suppose that $(\alpha, \delta)-\mathsf{bSVP}$ holds for every constants $\delta \in (0, 1/4), \alpha \in (0, 1)$ which satisfy $\delta < \alpha/2$. Then, for every constants $d > 4$ and $\gamma > 4/d$ there exists a degree-d CRH with shrinkage factor of γ.

For example, we can get a degree 5 CRH with shrinkage-factor of 0.81 or degree-8 CRH with shrinkage factor of 0.51.

Proof. Let $\beta = 2^{-d}$ and recall that item 2 of Lemma 13 provides a (b, β)-expanding mapping Expand' with $b = \frac{1}{\beta \log(1/\beta)} = 2^d/d$ and output locality (and therefore also degree) of $\log(1/\beta) = d$. Let $\delta = 2\beta = 2^{-d+1}$ and $\alpha = \gamma/b$. Since $\gamma > 4/d$, it follows that $\delta < \alpha/2$, which, according to our assumption, implies that $(\alpha, \delta) - \mathsf{bSVP}$ holds. By plugging Expand' and the matrix-sampler that samples uniform $\alpha n \times n$ matrices into Construction 11, we get, by Lemma 12, a degree-d CRH with shrinkage factor of γ. The proposition follows. ◀

▶ Remark. Recall that we measure the degree of a collection of functions $H = \{h_z\}$ as the maximal degree of each function in the collection and ignore the degree of the evaluation algorithm H which maps the collection key z and the input x to $h_z(x)$. (See Remark 2.) Nevertheless, it is not hard to see that all the constructions of this section admit an evaluation algorithm of constant degree. (In fact, the degree is $d + 1$ where d is the degree of h_z).

4.3 Locally-Computable CRH

Our next goal is to construct CRH's with constant output and input locality. To this end, we instantiate bSVP with the LDPC ensemble.

▶ **Theorem 19.** *Suppose that there exist constants $\delta \in (0, 1/4), \alpha \in (0, 1)$ with $\delta > 2H_2^{-1}(\alpha)$ and a constant $d \in \mathbb{N}$ for which $(d, \alpha, \delta) - \mathsf{bSVP}$ holds. Then, there exists a linearly-shrinking CRH with constant input locality and constant output locality. Moreover, one can reduce the output locality to 4 (while keeping the shrinkage linear and the input locality constant).*

Proof. Fix δ and α and take $\beta = \delta/2$ and $b \in (1/H_2(\beta), 1/\alpha)$. Let Expand be the (b, β)-expanding mapping promised in item 1 of Lemma 13 which has input locality of c' and output locality d'. Instantiate Construction 11 with Expand and with the LDPC matrix sampler $\mathcal{M}$ which samples a uniformly chosen $\alpha \cdot n \times n$ matrix subject to the constraint that each column contains exactly $c = \alpha d$ ones and each row contains exactly d ones. Then, by Lemma 12, we derive CRH $\mathcal{H} = \{h_{\mathbf{M}}\}$ with linear shrinkage, output locality of $D = d \cdot d'$ and input locality of $C = c \cdot c'$. This proves the first part of the theorem.

For the second part, take the CRH $\mathcal{H} = \left\{h_z : \{0,1\}^k \to \{0,1\}^m\right\}$ constructed in the first part of the theorem, and apply the 4-local perfect encoding promised in Proposition 4. The resulting collection $\hat{\mathcal{H}}$ has the required syntactic properties, and, by Lemma 3, it forms a CRH. ◀

Since any NC^0 function can be computed by a circuit of linear size, Theorem 14 yields a linear-time computable CRH with linear-shrinkage. Such a function can be turned into a linear-time computable CRH with arbitrary polynomial-stretch (using a hash-tree), we therefore derive the following corollary.

▶ **Corollary 20.** *Under the assumption of Theorem 19, for every constant $c < 1$ there exists a CRH $\mathcal{H} = \left\{h_z : \{0,1\}^k \to \{0,1\}^{k^c}\right\}$ which can be computed by a circuit of size $O(k)$.*

5 Degree-2 Hash Functions

In Section 5.1, we construct a universal one-way hash function family under the MQ assumption (see Definition 10 for the definition of the $\mathcal{D}_{n,m,p}$-MQ assumption). We then turn our attention to collision-resistance. In Section 5.2, we show that a natural family of uniformly random quadratic functions is *not* collision-resistant, by showing an explicit polynomial-time attack. An intriguing question left open by our work is the existence of a degree-2 CRH family. In Section 5.3, we show how this question (and a related question on statistically hiding commitments) relates to a long-standing question on constructing degree-2 randomized encodings.

5.1 Universal One-way Hash Function

▶ **Construction 21.** Let λ be the security parameter, and let $n = n(\lambda)$, $m = m(\lambda)$, $p = p(\lambda)$ (with $m < n$) be the MQ parameters. Let the MQ distribution $\mathcal{D} = \mathcal{D}_{n,m,p}$ be the uniform distribution that outputs a set of m uniformly random upper-triangular matrices $\mathbf{Q}_i$ and m vectors $\mathbf{L}_i$.

Define the family of hash functions $\mathcal{H}_{n,m,p} = \{h_{\vec{\mathbf{Q}},\vec{\mathbf{L}}} : \mathbb{Z}_p^n \to \mathbb{Z}_p^m \ : \ \vec{\mathbf{Q}} \in (\mathbb{Z}_p^{n\times n})^m,\ \vec{\mathbf{L}} \in (\mathbb{Z}_p^n)^m\}$ as follows:

$$h_{\vec{\mathbf{Q}},\vec{\mathbf{L}}}(\mathbf{x}) = \left(\mathbf{x}^T\mathbf{Q}_i\mathbf{x} + \mathbf{L}_i^T\mathbf{x}\right)_{i=1}^{m}$$

We now show that under the MQ assumption, this family is universal one-way.

▶ **Theorem 22.** *Let $n = n(\lambda)$, $m = m(\lambda)$, $p = p(\lambda)$ and $\mathcal{D}_{n,m,p}$ be such that (a) $m < n$ and (b) $\mathcal{D}_{n,m,p}$ is the uniform distribution. Then, under the $\mathcal{D}_{n,m,p}$-MQ assumption, the family $\mathcal{H}_{n,m,p}$ is universal one-way.*

The construction and proof immediately generalize to other families of distributions where the distribution of $\mathbf{Q}_i$ is arbitrary but $\mathbf{L}_i$ is still uniformly random. In this exposition, we choose to present the simpler version where both $\mathbf{Q}_i$ and $\mathbf{L}_i$ are uniformly random.

Proof. First, since $m < n$, $\mathcal{H}_{n,m,p}$ is a compressing family of functions.

We now show that it is universally one-way. Assume that there is a PPT UOWHF-breaker algorithm $(\mathsf{Adv}_1, \mathsf{Adv}_2)$. We will construct an algorithm $\mathcal{B}$ that breaks the $\mathcal{D}$-MQ assumption. $\mathcal{B}$ gets as input an MQ challenge $(\mathbf{Q}_i, \mathbf{L}_i)_{i=1}^m$ and does the following.

- Run Adv_1 to get a target input $\mathbf{x}$ and state information r.
- Define the UOWHF family using the matrices $\mathbf{Q}_i^*$ and $\mathbf{L}_i^*$ where:

$$\mathbf{Q}_i^* = \mathbf{Q}_i \text{ and } \mathbf{L}_i^* = \mathbf{L}_i - (\mathbf{Q}_i^* + (\mathbf{Q}_i^*)^T)\mathbf{x} \tag{5.1}$$

- Feed $\vec{\mathbf{Q}} := (\mathbf{Q}_1^*, \ldots, \mathbf{Q}_m^*)$ and $\vec{\mathbf{L}} := (\mathbf{L}_1^*, \ldots, \mathbf{L}_m^*)$ to Adv_2 together with the state information r. Get back a colliding input $\mathbf{y}$, output $\mathbf{y} - \mathbf{x}$ and halt.

First note that the distribution of $\mathbf{Q}_i$ and $\mathbf{L}_i$ from equation 5.1 are uniformly random and hence, the adversary $\mathsf{Adv} = (\mathsf{Adv}_1, \mathsf{Adv}_2)$ will find a colliding input $\mathbf{y}$ with non-negligible probability $1/q(\lambda)$.

Let $\Delta := \mathbf{y} - \mathbf{x}$. Since $\mathbf{x}$ and $\mathbf{y}$ are colliding inputs, we have

$$(\mathbf{x}+\Delta)^T\mathbf{Q}_i^*(\mathbf{x}+\Delta) + (\mathbf{L}_i^*)^T(\mathbf{x}+\Delta) = \mathbf{x}^T\mathbf{Q}_i^*\mathbf{x} + (\mathbf{L}_i^*)^T\mathbf{x}$$

for all $i \in [m]$. A quick calculation then tells us that

$$\Delta^T\mathbf{Q}_i\Delta + \Delta^T\mathbf{Q}_i\mathbf{x} + \mathbf{x}^T\mathbf{Q}_i\Delta + (\mathbf{L}_i^*)^T\Delta = \Delta^T\mathbf{Q}_i\Delta + \left(\mathbf{x}^T\mathbf{Q}_i^T + \mathbf{x}^T\mathbf{Q}_i + (\mathbf{L}_i^*)^T\right)\Delta = 0$$

which in turn gives us

$$\Delta^T\mathbf{Q}_i\Delta + \mathbf{L}_i^T\Delta = 0 \ ,$$

by our definition of $\mathbf{L}_i := \mathbf{L}_i^* + (\mathbf{Q}_i^* + (\mathbf{Q}^*)^T)\mathbf{x}$. Thus, $\mathcal{B}$ outputs a solution to the challenge MQ instance with probability $1/p(\lambda)$ as well, which shows that $\mathcal{H}_{n,m,p}$ is a universal one-way hash family. ◀

Generalizing to Other MQ Distributions

Our proof readily generalizes to distributions for the MQ problem which output an arbitrary distribution of the quadratic forms $\mathbf{Q}_i$ and a uniformly random distribution of the linear forms $\mathbf{L}_i$.

Generalizing to Larger Degrees

Our construction and proof also generalize to the setting where the hash function has degree d and can be based on the hardness of solving random degree-d polynomial equations, a generalization of the MQ assumption. The reason for considering this generalization is to achieve better shrinkage. The MQ construction can shrink n bits to no less than $m = \sqrt{n}$ bits, since the MQ assumption is false for smaller m. Since we do not know attacks against the degree-d assumption with shrinkage $m = n^{\Omega(1/d)}$, this variant will give us larger shrinkage at the expense of larger degree (and a different assumption).

5.2 Finding Collisions in Random Degree-2 Functions

We now show that the hash function family $\mathcal{H}_{n,m,p}$ is *not* collision-resistant. We remind the reader that $\mathcal{H}_{n,m,p}$ refers to the function where $\mathbf{Q}_i$ are uniformly random upper-triangular matrices and $\mathbf{L}_i$ are uniformly random vectors. The attack we describe below was discovered by Ding and Yang [30] but without a proof of correctness.

▶ **Theorem 23.** *For every $n, m < n$ and p, there is a* $\mathsf{poly}(n, m, \log p)$*-time algorithm* ColFinder *such that*

$$\Pr_{h \leftarrow \mathcal{H}_{n,m,p}}[\mathsf{ColFinder}(h) = (\mathbf{x}, \mathbf{y}) : \ h(\mathbf{x}) = h(\mathbf{y}) \wedge \mathbf{x} \neq \mathbf{y}] = \Omega(1)$$

In other words, the family $\mathcal{H}_{n,m,p}$ is not collision-resistant.

Proof. The strategy of ColFinder is simple. It chooses a uniformly random $\Delta \in \mathbb{Z}_p^n$ and solves the system of equations

$$\left\{ \ (\mathbf{x}+\Delta)^T\mathbf{Q}_i(\mathbf{x}+\Delta) + \mathbf{L}_i^T(\mathbf{x}+\Delta) = \mathbf{x}^T\mathbf{Q}_i\mathbf{x} + \mathbf{L}_i^T\mathbf{x} \ \right\}_{i\in[m]}$$

This is in fact a linear system of equations in the unknown $\mathbf{x}$:

$$\left\{ \ \mathcal{L}_i : \mathbf{x}^T(\mathbf{Q}_i^T + \mathbf{Q}_i)\Delta + \mathbf{L}_i^T\Delta = 0 \ \right\}_{i\in[m]}$$

where all computations are done mod p. Any solution $\mathbf{x}$ to this system of linear equations gives us a collision $(\mathbf{x}, \mathbf{x} + \Delta)$. Conversely, if there exists a collision with difference Δ, ColFinder will find such a collision.

It remains to show that for uniformly random upper-triangular matrices $\mathbf{Q}_i$ (and possibly uniformly random Δ and $\mathbf{L}_i$), this system has a solution with high probability. We show a stronger statement: namely, that the n-by-m matrix $\widetilde{\mathbf{Q}}$ defined as

$$\widetilde{\mathbf{Q}} = \begin{bmatrix} | & \vdots & | \\ (\mathbf{Q}_1^T + \mathbf{Q}_1)\Delta & \vdots & (\mathbf{Q}_m^T + \mathbf{Q}_m)\Delta \\ | & \vdots & | \end{bmatrix}$$

has full rank, namely rank m a constant probability. This in turn implies that the equations $\mathcal{L}_i$ are guaranteed to have a solution with constant probability.

Let us now bound the probability that $\widetilde{\mathbf{Q}}$ has rank less than m. To do so, let us understand the distribution of $\widetilde{\mathbf{Q}}$ using the following observations.

- Fix an $i \in [m]$ be such that $\Delta_i \neq 0$. Such an i is guaranteed to exist as $\Delta \neq \mathbf{0}$.
  We now claim that for each $j \in [m]$, the entries of $(\mathbf{Q}_j + \mathbf{Q}_j^T)\Delta$ are uniformly random except possibly for the i-th entry. This follows from the fact that the i-th column of $(\mathbf{Q}_j + \mathbf{Q}_j^T)$ is uniformly random (except for its i-th entry) and uncorrelated to the rest of the matrix except the i-th row (which is identical to the i-th column by symmetry).
- This implies, in turn, that $\widetilde{\mathbf{Q}}$ has a uniformly random $(n-1)$-by-m submatrix. The probability that this is full-rank, namely rank $\mathsf{min}(n-1, m) = m$, is a constant (see, e.g., [17]).

Put together, we see that ColFinder succeeds with constant probability in finding a collision. We remark that this argument did not use the randomness of $\mathbf{L}_i$ or Δ, but rather only the fact that the $\mathbf{Q}_i$ are uniformly random upper-triangular matrices. ◀

5.3 Degree-2 CRH via Randomized Encoding?

An intriguing question left open by our work is the existence of a degree-2 CRH. The same question is open also for the related primitive of non-interactive statistically hiding commitments (SHC), which can be easily constructed from a CRH (see Appendix A). We relate these questions to questions about the existence of degree-2 statistically private randomized encodings that were left open by [51].

We start by defining a relaxation of the perfect notion of randomized encoding from Definition 2 that allows for statistical privacy error and eliminates the balanced simulation and length requirements.

▶ **Definition 24** (Statistically-Private, Perfectly Correct Randomized Encoding)**.** Let $f : \{0,1\}^n \to \{0,1\}^m$ be a function. We say that a function $\hat{f} : \{0,1\}^n \times \{0,1\}^\rho \to \{0,1\}^s$ is an *ϵ-private, perfectly correct randomized encoding* (ϵ-RE) of f if there exists a deterministic decoding algorithm C and a randomized simulator S which satisfy the following:

- (Perfect correctness.) For every input $x \in \{0,1\}^n$ and $r \in \{0,1\}^\rho$, it holds that $C(\hat{f}(x;r)) = f(x)$.
- (ϵ-privacy) For every $x \in \{0,1\}^n$, the distribution $\hat{f}(x;r)$, induced by a uniform choice of $r \xleftarrow{R} \{0,1\}^\rho$, satisfies $\mathsf{SD}(\hat{f}(x;r), S(f(x))) \leq \epsilon$, where SD denotes statistical distance.

For the dual notion of ϵ-correct, perfectly-private RE, it is shown in [51] that only very special functions admit such an encoding with a degree-2 $\hat{f}$ (in the Boolean case, this class

of functions includes only degree-2 polynomials in the input and functions that test whether the input is in an affine subspace of $\{0,1\}^n$). It is open whether the same holds for the above notion of ϵ-RE.

▶ **Question 25.** Does every finite $f : \{0,1\}^n \to \{0,1\}$ admit a family of degree-2, ϵ_t-private, perfectly correct randomized encodings $\hat{f}_t : \{0,1\}^n \times \{0,1\}^{\rho_t} \to \{0,1\}^{s_t}$ with $\epsilon_t = \text{neg}(t)$ and $\rho_t, s_t = \text{poly}(t)$?

This question is open even for the specific function $f : \{0,1\}^4 \to \{0,1\}$ defined by $f(a,b,c,d) = abc \oplus d$ and even with fixed ϵ (say $\epsilon = 1/3$). In fact, it follows from [1] that the latter function is complete in the sense that an affirmative answer to Question 25 for this function implies an affirmative answer for all functions.

We show that an affirmative answer to Question 25 together with standard cryptographic assumptions would imply the existence of a degree-2 SHC. Thus, ruling out such an SHC would effectively require settling Question 25 in the negative.

A (non-interactive) SHC is defined by a collection $C_z(x,\sigma)$, which given a public random string z (that can be reused for many commitments) maps an input x and secret randomness σ to a commitment c. The commitment c should statistically hide x. It should also be *computationally binding* in the sense that given z it is infeasible to find a pair (x,σ) and (x',σ') which are consistent with the same c, where $x \neq x'$. See Appendix A for a formal definition.

▶ **Theorem 26.** *Suppose there is a CRH or SHC in* NC^1*. Moreover, suppose that the answer to Question 25 is affirmative. Then there is a degree-2 SHC.*

Proof. Using Theorem 37, a CRH in NC^1 implies an SHC in NC^1, which in turn implies an SHC $C_z(x,\sigma)$ in NC^0 [1]. Since every output bit of $C_z(x,\sigma)$ depends on a constant number of input bits, we can apply the degree-2 encoding implied by an affirmative answer to Question 25, independently to every output bit of C_z and with $t = |x|$, viewing it as a deterministic function of (x,σ). This yields a polynomial-size degree-2 function $\hat{C}_z(x,\sigma;\tau)$ which is an ϵ-RE of C_z with ϵ negligible in $k = |x|$. We view $\hat{C}_z$ as a commitment scheme with input x and secret randomness (σ,τ). The perfect correctness requirement of the ϵ-RE implies that $\hat{C}_z$ is computationally binding (since if $(x,(\sigma,\tau))$ and $(x',(\sigma',\tau'))$ violate the binding of $\hat{C}_z$ then (x,σ) and (x',σ') violate the binding of C_z). The statistical hiding property is implied by the fact that the error ϵ of the ϵ-RE is negligible in the input length. Indeed, for every $x \neq x'$ we have

$$\hat{C}_z(x;\sigma,\tau) \approx S_z(C_z(x;\sigma)) \approx S_z(C_z(x';\sigma)) \approx \hat{C}_z(x';\sigma,\tau),$$

where σ,τ are uniformly distributed, $\approx$ denotes statistical indistinguishability, and S_z is the simulator of the encoding. ◀

For the case of CRH, even an affirmative answer to Question 25 does not seem to suffice for a degree-2 construction, since an ϵ-RE of a CRH may lose the shrinking property. Instead, we formulate an ad-hoc variant of randomized encoding that captures a minimal set of requirements needed for respecting the CRH properties.

▶ **Definition 27** (CRH-Respecting Randomized Encoding)**.** Let $f : \{0,1\}^n \to \{0,1\}^m$ be a function with $m < n$. We say that a function $\hat{f} : \{0,1\}^n \times \{0,1\}^\rho \to \{0,1\}^s$ is a *CRH-respecting randomized encoding* of f if the following hold:

- (Perfect correctness.) There exists a deterministic decoding algorithm C such that for every input $x \in \{0,1\}^n$ and $r \in \{0,1\}^\rho$, it holds that $C(\hat{f}(x;r)) = f(x)$.

- (Injective randomness) For any fixed x, the function $\hat{f}(x;\cdot)$ is injective; namely for any $r \neq r'$ we have $\hat{f}(x;r) \neq \hat{f}(x;r')$.
- (Shrinkage) The encoding $\hat{f}(x;r)$ is shrinking, namely $s < n + \rho$.

Note that given the perfect correctness and injective randomness requirements, the best one can hope for is to match the shrinkage of f, namely the shortest possible encoding output is of size $s = \rho + m$. If the shrinkage requirement of Definition 27 is strengthened to require this optimal shrinkage, the definition becomes equivalent to the notion of PRE from Definition 2 [1]. In particular, perfect privacy is implied by the perfect correctness, injective randomness, and optimal shrinkage.

On the other hand, even the relaxed requirements of Definition 27 *do* imply some weak form of "average-case privacy." Indeed, if the input x could always be recovered from $\hat{f}(x,r)$, then injective randomness guarantees that r can also be recovered, and since $s < n + \rho$ we get a contradiction. Thus, the notion of CRH-respecting randomized encoding crudely has the same flavor of perfect correctness and partial privacy as the notion of ϵ-RE from Definition 24.

We now show that applying a CRH-respecting randomized encoding to a CRH indeed yields a CRH. (We use concrete function notation instead of infinite collections of functions for simplicity.)

▶ **Claim 1.** *Suppose $h_z : \{0,1\}^n \to \{0,1\}^m$ is a CRH and $\hat{h}_z : \{0,1\}^n \times \{0,1\}^\rho \to \{0,1\}^s$ is an efficient, CRH-respecting randomized encoding of h_z. Then the function $\hat{h}_z$, viewed as a mapping from $n + \rho$ input bits to s output bits, is a CRH.*

Proof. First, the shrinkage requirement directly guarantees that $\hat{h}$ is shrinking its input (x,r). We show that $\hat{h}$ inherits the collision resistance of h. Suppose that $\mathsf{Adv}(z)$, given a random z, finds a collision $(x,r),(x,r')$ for $\hat{h}_z$, where $(x,r) \neq (x',r')$. By the injective randomness requirement we must have $x \neq x'$ and by perfect correctness we must have $h(x) = h(x')$. Hence, Adv can be used to find a collision (x,x') for h with the same success probability. ◀

Finally, we pose a concrete question about the existence of CRH-respecting randomized encodings which is related to the existence of degree-2 CRH.

▶ **Question 28.** Does every $f : \{0,1\}^n \to \{0,1\}^{\lfloor n/10 \rfloor}$ in NC^0 admit (an efficiently computable) degree-2 CRH-respecting randomized encoding?

Under the assumptions of Theorem 19, ruling out a degree-2 CRH would require proving a negative answer to Question 28.

6 Win-Win Results

In this section we prove that a failure of our constructions leads to interesting algorithmic consequences. Both of our results are based on the following observation.

▶ **Lemma 29.** *Suppose that there exists an algorithm Adv with complexity T for which*

$$\Pr_{\mathbf{M} \xleftarrow{R} \mathbb{Z}_2^{m \times n}} [\mathsf{Adv}(\mathbf{M}) = \mathbf{x} \text{ such that } \mathbf{x} \neq \mathbf{0}, \mathbf{M}\mathbf{x} = \mathbf{0} \text{ and } \Delta(\mathbf{x}) \leq \delta] \geq \epsilon.$$

Then there exists an algorithm Adv' with complexity $T' = T + \mathrm{poly}(m)$ such that for every $\mathbf{y} \in \mathbb{Z}_2^m$ it holds

$$\Pr_{\mathbf{M} \xleftarrow{R} \mathbb{Z}_2^{m \times n}} [\mathsf{Adv}'(\mathbf{M}, \mathbf{y}) = \mathbf{x} \text{ such that } \mathbf{M}\mathbf{x} = \mathbf{y} \text{ and } \Delta(\mathbf{x}) \leq \delta] \geq \epsilon/2.$$

That is, an algorithm Adv that finds a small subset of the columns of $\mathbf{M}$ that spans the all-zero vector, can be transformed into an algorithm Adv′ that finds a small subset of the columns that spans *any* given target vector $\mathbf{y}$.

Proof. The algorithm Adv′ is given a random matrix $\mathbf{M} \stackrel{R}{\leftarrow} \mathbb{Z}_2^{m\times n}$ and an arbitrary target vector $\mathbf{y} \in \mathbb{Z}_2^m$. It samples a vector $\mathbf{u} \stackrel{R}{\leftarrow} \mathbb{Z}_2^n$ and calls Adv with the input $\mathbf{M}' = \mathbf{M} + \mathbf{y} \cdot \mathbf{u}^T$ (note that this is an outer-product). Note that if the output $\mathbf{x}$ of Adv is (1) a valid solution (i.e., it is a non-zero vector of weight at most δn and is in the Kernel of $\mathbf{M}$); and (2) is non-orthogonal to $\mathbf{u}$ (i.e., $\langle \mathbf{x}, \mathbf{u}\rangle = 1$), then $\mathbf{Mx} = \mathbf{Mx} + (\mathbf{y} \cdot \mathbf{u}^T)\mathbf{x} = \mathbf{y}$ and so Adv′ succeeds.

To analyze the success probability, note that the joint distribution of $(\mathbf{M}, \mathbf{u})$ is uniform and therefore (1) happens with probability ϵ, by assumption. Moreover, conditioned on (1), the probability of (2) is exactly $\frac{1}{2}$. The lemma follows. ◀

6.1 bSVP and Distributed Parity-Learning

In this section we show that if the bSVP assumption (as phrased in Corollary 17) does not hold, one can learn parities in a distributed setting with non-trivial memory/communication tradeoffs. The influence of memory and communication restrictions on learning in distributed environments has been studied recently by several works (cf. [6, 68, 70, 67] and references therein). We consider here another variant of this question.

Let us first recall the standard notion of PAC-learning parity functions over uniformly sampled examples. For a secret parity function f_s (s is an m-bit vector), the learner is given random (non-noisy) labeled samples of the form (r_i, b_i) where $r_i \stackrel{R}{\leftarrow} \mathbb{Z}_2^m$ is a random example and $b_i = f_s(r_i) = \langle r_i, s\rangle$ is a binary label. The goal is to predict f_s on a random vector $r^* \stackrel{R}{\leftarrow} \mathbb{Z}_2^m$ with probability, say, 2/3. We consider the case where the learner is composed of two parties: a measurement device W that collects the samples and has only limited memory of S bits and no computational power, and an analyst A who runs in polynomial-time and, given r^*, attempts to predict $f_s(r^*)$. We assume that A sees the vectors r_i's but can access a bit b_i only by reading it from W's memory.[6] Our goal is to minimize the number of bit probes that A makes (subject to the given memory bound S).

In the following we think of a memory-bound of $S = cm$ for some $c > 1$. In this case, W can store cm labels $(b_1, \ldots, b_S)$ and A can trivially achieve a communication of m bits by finding an m-size subset $R' \subset R \stackrel{\Delta}{=} \{r_1, \ldots, r_m\}$ of linearly-independent examples and ask for the corresponding labels. At this point A can recover s and compute $f_s(r^*)$. In fact, A can reduce the communication to $m/2$: first write r^* as a linear combination $\sum v_i R'_i$ of the vectors in R', then ask only for the b_i for which $v_i \neq 0$, and finally compute $f_s(r^*)$ by $\sum v_i b_i$. It is not hard to show (similarly to Observation 8) that the expected communication is $m/2$. More generally, the analyst A can achieve a communication of w bits if she can write the challenge vector r^* as a w-weight linear combination of the example vectors R. As shown in Lemma 29, this problem reduces to the bSVP problem. In particular, we prove the following lemma.

[6] This captures a scenario where the inputs r_i are public (e.g., generated by some environment) but only the measurement device gets to see how the function reacts to it and measure $f_s(r_i)$. We mention that the results of this section carry over (with minor adaptations) to a setting where the r_i's are only given to W.

▶ **Lemma 30.** *Given an efficient algorithm B that solves the $(\alpha, \delta) - \mathsf{bSVP}$ problem with probability $2/3$ there exists an efficient algorithm that solves the distributed parity-learning problem with memory limitation of m/α and with $m(\delta/\alpha)$ bit probes.*

Proof. The analyst A is given a random matrix of examples $R \xleftarrow{R} \mathbb{Z}_2^{m \times S}$, where $S = m/\alpha$, and a random challenge $r^* \xleftarrow{R} \mathbb{Z}_2^m$. It calls the algorithm B' promised in Lemma 29 with input R and target vector r^*. If the algorithm succeeds (which happens with probability $1/3$), the analyst gets a vector v of weight δS for which $Rv = r^*$, and can predict $f_s(r^*)$ using $\delta S = m(\delta/\alpha)$ bit probes. Otherwise, the analysis outputs a random bit. The overall success probability is $\geq 1/3 + 2/3 \cdot (1/2) = 2/3$, as required. ◀

Recall that in Corollary 17 we showed that if there exist constants $\delta \in (0, 1/4), \alpha \in (0, 1)$ with $\delta > 2H_2^{-1}(\alpha)$ for which $(\alpha, \delta) - \mathsf{bSVP}$ holds, then degree-3 linearly-shrinking CRH exist. Using standard amplification techniques, one can prove a similar result even under the assumption that $(\alpha, \delta) - \mathsf{bSVP}$ cannot be solved in polynomial time with probability better than $2/3$.[7] Overall, by combining this with Lemma 30 and the approximation of the inverse entropy function from Eq. (2.2), we conclude the following "win-win" result.

▶ **Corollary 31.** *At least one of the following holds:*

- *There exists a degree-3 linearly-shrinking CRH.*
- *For any constant $c > 1$ and any $\gamma > (2/\log c)$, the distributed parity problem can be solved efficiently with memory-bound of cm and γm bit probes for infinitely many m.*

Note that the bit probe rate γ tends to zero when c grows – we are not aware of any efficient solution which achieves such a dependency.

6.2 Speeding-up the BKW algorithm

We move on to the more traditional setting of *learning parity with noise* (LPN) [18]. Recall that in this setting the learner is given random noisy labeled samples of the form (r_i, b_i) where $r_i \xleftarrow{R} \mathbb{Z}_2^m$ is a random example and $b_i = f_s(r_i) + \chi_i$ is a binary label where f_s is a parity function (specified by a secret $s \in \{0,1\}^m$) and χ_i is a random variable which takes the value 1 with probability τ for some noise parameter $\tau \in (0, \frac{1}{2})$. The learner should be able to recover s with, say, probability $2/3$ while minimizing the running time and the sample complexity.[8]

The best known algorithm for solving the LPN problem (i.e., to recover s), due to Blum, Kalai and Wasserman [19], runs in time (and sample) complexity of $2^{O(m/\log m)}$. We show that either the complexity can be reduced to $2^{cm/\log m}$ for arbitrary small constant $c > 0$, or linearly-shrinking CRH of logarithmic degree exist. To this end, we consider the bSVP assumption (for random linear codes) in the polynomial regime, i.e., when the number of columns n is polynomially larger than the number of rows m.

---

[7] Indeed, by plugging the weaker assumption in our construction, we get a linearly-shrinking degree-3 "weak"-CRH in which the probability of finding collisions (as defined in Eq. (2.3)) is at most $2/3$ as opposed to negligible. Such a CRH can be amplified into standard CRH while preserving the degree and the linear-shrinkage by expanding the k-bit input x into an $O(k)$-bit vector y via a linear (fixed) error correcting code and then shrinking y down to $(1-\epsilon)k$-long vector, by applying independent copies of the weak CRH to distinct blocks of size $\sqrt{k}$, see [25] for further details. (In fact, one can even get a locality-preserving transformation [3, Lemma 5.7].)

[8] Again, we could define the goal as predicting the value $f_s(r^*)$ for random r^* with non-trivial success probability. However, in this setting the two goals reduces to each other with polynomial overhead.

▶ **Assumption 32.** There exists a positive constant $a > 1$ for which (α, δ)-bSVP holds for $\alpha = 1/n^{1/a}$ and $\delta = 8\alpha/(\log(1/\alpha))$.

The constant 8 in the above is somewhat arbitrary and any constant larger than 2 suffices. It will be useful to state the above assumption in terms of the parameter m (and not n as usual). That is, the assumption asserts that, given a random $m \times (n = m^a)$ binary matrix $\mathbf{M}$, it is hard to find a vector of weight $w = \delta n = \frac{8m}{(a-1)\log m}$ in the kernel of M. Note that, unlike in the previous sections, now the dimensions of the matrix are polynomially related (and not linear) and correspondingly we can ask for a kernel vector of sub-linear weight.

▶ **Lemma 33.** *Suppose that Assumption 32 does not hold. Then, for every constant $c > 0$ and constant noise rate $\tau \in (0, \frac{1}{2})$ there exists an algorithm that for infinitely many m's, solves the m-dimensional LPN problem with noise rate τ (i.e., recovers the secret s) in time (and sample complexity) of $N = poly(m) \cdot 2^{cm/\log m}$.*

Proof. We describe an algorithm that in time N computes s_1, the first bit of s, with probability 2/3. Using standard amplification, and by exploiting the symmetry of the LPN problem, such an algorithm can be converted to an algorithm that recovers s with, say, probability 2/3 at the expense of increasing the time and sample complexity by a factor of $q(m)$ for some fixed (universal) polynomial $q(\cdot)$. (First reduce the error below $1/3m$ via repetition and majority vote, and then apply the amplified algorithm m times where in each iteration we recover the i-bit of s by rotating the examples $i-1$ coordinates to the left.)

Let $a > 1$ be a constant whose value will be determined later, and let $w = \frac{8m}{(a-1)\log m}$. Since Assumption 32 does not hold, there exists, by Lemma 29, a polynomial-time algorithm Adv_a such that for infinitely many m's, for all $\mathbf{y} \in \mathbb{Z}_2^m$

$$\Pr_{\mathbf{M} \xleftarrow{R} \mathbb{Z}_2^{m \times m^a}} [\mathsf{Adv}_a(\mathbf{M}, \mathbf{y}) = \mathbf{x} \text{ such that } \mathbf{Mx} = \mathbf{y} \text{ and } \mathbf{x} \text{ has Hamming weight of at most } w],$$

is larger than some inverse polynomial $\epsilon(m)$.

Let $n = m^a$, $\mu = 1 - 2\tau$, $t = O(\mu^{2w})$ and $t' = O(t/\epsilon)$. The algorithm asks for nt' labeled examples and partitions them into t' sets T_i each of n examples. For each set T_i, let S_i denote the set of examples without their noisy labels. We apply Adv_a to S_i and set the target vector $\mathbf{y}$ to be the first unit vector $\mathbf{e}_1$. If Adv_a succeeds (i.e., returns a subset $S_i' \subseteq S_i$ of size $\frac{bm}{\log m}$ whose sum is $\mathbf{e}_1$) then we XOR together the labels that correspond to the vectors in S_i' and record the result as a vote v_j (which serves as a guess for s_1). If the algorithm fails, we do not record the vote. Finally, we output the majority of all recorded votes v_j.

Analysis: First we claim that each recorded vote is correct (equals to s_1) independently with probability $1/2 + \mu^w$. Indeed, each vote is of the form $s_1 + \sum_{i=1}^{w} \chi_i$ where the χ_i's are independent Bernoulli variables each with expectation τ. It is well known (e.g., [19, Lemma 4]) that $\chi = \sum_{i=1}^{w} \chi_i$ is a Bernoulli random variable with expectation of $(1 - \mu^w)/2$ for $\mu = 1 - 2\tau$.

Next, we argue that, except with probability 1/10, there is a large number of votes. Indeed, each invocation of Adv_a succeeds with at least ϵ probability hence, by Markov's inequality, the probability that there are less than t successful invocations out of, say $t' = 10t/\epsilon$ attempts, is at most 1/10.

Finally, conditioned on having t votes, by a Chernoff bound, the probability that the final outcome is wrong is 1/10. The claim follows by a union bound.

Overall, the time complexity of the algorithm is

$$O(nt') = O(m^a \cdot (1-2\tau)^{\frac{20m}{(a-1)\log m}}) = O(m^a \cdot e^{\frac{8m}{\tau(a-1)\log m}}),$$

where e is the natural exponent. Hence, we get the desired running time by taking $a = a(\tau, c)$ to be a sufficiently large constant. ◀

Next we show that if Assumption 32 holds then we get CRH of logarithmic degree.

▶ **Lemma 34.** *Under Assumption 32, there exists a linearly-shrinking CRH with logarithmic degree.*

Proof. Let $a > 1$ be the constant promised by Assumption 32 and let $\alpha = \frac{1}{n^{1/a}}$, $\delta = \frac{8\alpha}{\log(1/\alpha)}$. Also, let

$$d = \log(2/\delta) = \log(n^{1/a}) + \log\log(n^{1/a}) - 2, \quad \beta = 2^{-d},$$

$$b = \frac{2^d}{d} = \frac{n^{1/a}\log(n^{1/a})}{4(\log(n^{1/a}) + \log\log(n^{1/a}) - 2)}.$$

We instantiate Construction 11 with the (α, δ)-bSVP sampler and with the (b, β)-expanding mapping promised in the second item of Lemma 13. (This part of the lemma holds for non-constant β as well.) Since $\beta \leq \delta/2$ and $b\alpha < 1/4$, we get, by Lemma 12, a linearly shrinking CRH with degree of $d = O(\log n)$. The lemma follows. ◀

We conclude the following corollary.

▶ **Corollary 35.** *At least one of the following holds:*

- *There exists a linearly-shrinking CRH with logarithmic degree.*
- *For any constant $c > 0$ and any constant noise rate $\tau \in (0, \frac{1}{2})$ there exists an algorithm that, for infinitely many m, solves the m-dimensional LPN problem with noise rate τ in time and sample complexity of $poly(m) \cdot 2^{cm/\log m}$.*

7 Conclusions and Open Questions

Under plausible intractability assumptions, we establish the existence of low-complexity cryptographic hash functions that compress the input by (at least) a constant factor. In particular, we construct CRH with linear circuit size, constant locality, or algebraic degree 3 over $\mathbb{Z}_2$ under different flavors of the newly introduced binary SVP (bSVP) assumption. We also establish connections with other problems that either support our assumptions or indicate that further progress may be difficult.

While we provide some evidence supporting the validity of the flavors of bSVP we rely on, including a weak connection with the LPN problem, it is left open to obtain a better understanding of the relation between bSVP and LPN or other well studied cryptographic assumptions. It would also be interesting to obtain similar positive results under better or incomparable assumptions, such as the MQ assumption (that we use to construct degree-2 UOWHFs) or the one-wayness of random local functions (used in [3] for constructing local UOWHFs).

Our work leaves open several other natural questions. One such question is the existence of CRH (or even 2-message statistically hiding commitments) with degree 2 or output locality 3. Another is the maximal achievable compression of a degree-d CRH: The bSVP-based security analysis of our construction can only support a constant compression factor, which seems unlikely to be optimal. (In contrast, for *linear-size* CRH we can provide an arbitrary polynomial compression.) A final question is to understand the collision resistance properties of *random* degree-d mappings. While we rule out collision resistance for $d = 2$ with any non-trivial compression, the question is wide open for $d \geq 3$. It would be interesting to study the maximal compression (if any) for which a random degree-d mapping can be a CRH.

Acknowledgement. We thank Zvika Brakerski for participating in an earlier stage of this project. We also thank David Burstein, Simon Litsyn, Ronny Roth, Ohad Shamir, and David Woodruff for helpful discussions.

References

1 Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC0. *SIAM J. Comput.*, 36(4):845–888, 2006.

2 Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. How to garble arithmetic circuits. *SIAM J. Comput.*, 43(2):905–929, 2014.

3 Benny Applebaum and Yoni Moses. Locally computable UOWHF with linear shrinkage. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 486–502, 2013.

4 Daniel Augot, Matthieu Finiasz, and Nicolas Sendrier. A family of fast syndrome based cryptographic hash functions. In *Progress in Cryptology - Mycrypt 2005, First International Conference on Cryptology in Malaysia, Kuala Lumpur, Malaysia, September 28-30, 2005, Proceedings*, pages 64–83, 2005.

5 Jean-Philippe Aumasson and Willi Meier. Analysis of Multivariate Hash Functions. In *Information Security and Cryptology - ICISC 2007, 10th International Conference, Seoul, Korea, November 29-30, 2007, Proceedings*, pages 309–323, 2007.

6 Maria-Florina Balcan, Avrim Blum, Shai Fine, and Yishay Mansour. Distributed learning, communication complexity and privacy. In *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, pages 26.1–26.22, 2012.

7 Boaz Barak. How to go beyond the black-box simulation barrier. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 106–115, 2001.

8 Alexander Barg. Complexity issues in coding theory. *Electronic Colloquium on Computational Complexity (ECCC)*, 4(46), 1997.

9 Alexander Barg and G. David Forney Jr. Random codes: Minimum distances and error exponents. *IEEE Trans. Information Theory*, 48(9):2568–2573, 2002.

10 Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in 2 n/20: How 1 + 1 = 0 improves information set decoding. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, pages 520–536, 2012.

11 Mihir Bellare and Phillip Rogaway. Collision-resistant hashing: Towards making uowhfs practical. In *Advances in Cryptology - CRYPTO'97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, pages 470–484, 1997.

12 Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 1–10, 1988.

13 Côme Berbain, Henri Gilbert, and Jacques Patarin. QUAD: A multivariate stream cipher with provable security. *J. Symb. Comput.*, 44(12):1703–1723, 2009.

14 Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Smaller decoding exponents: Ball-collision decoding. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, pages 743–760, 2011.

15 Olivier Billet, Matthew J. B. Robshaw, and Thomas Peyrin. On Building Hash Functions from Multivariate Quadratic Equations. In *Information Security and Privacy, 12th Australasian Conference, ACISP 2007, Townsville, Australia, July 2-4, 2007, Proceedings*, pages 82–95, 2007.

16 Nir Bitansky, Ran Canetti, Alessandro Chiesa, Shafi Goldwasser, Huijia Lin, Aviad Rubinstein, and Eran Tromer. The hunting of the SNARK. *IACR Cryptology ePrint Archive*, 2014:580, 2014.

17 Johannes Blömer, Richard Karp, and Emo Welzl. The rank of sparse random matrices over finite fields. *Random Struct. Algorithms*, 10(4):407–419, 1997.

18 Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In *Advances in Cryptology - CRYPTO'93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, pages 278–291, 1993.

19 Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 435–440, 2000.

20 Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 112–117, 1982.

21 Elette Boyle, Niv Gilboa, and Yuval Ishai. Breaking the circuit size barrier for secure computation under DDH. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*, pages 509–539, 2016.

22 Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. *SIAM J. Comput.*, 43(2):831–871, 2014.

23 David Burshtein and Gadi Miller. Bounds on the performance of belief propagation decoding. *IEEE Trans. Information Theory*, 48(1):112–122, 2002.

24 Chris Calabro. *The Exponential Complexity of Satisfiability Problems*. PhD thesis, University of California, San Diego, 2009.

25 Ran Canetti, Ronald L. Rivest, Madhu Sudan, Luca Trevisan, Salil P. Vadhan, and Hoeteck Wee. Amplifying collision resistance: A complexity-theoretic treatment. In *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, pages 264–283, 2007.

26 David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 11–19, 1988.

27 Ronald Cramer, Ivan Damgård, and Ueli M. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, pages 316–334, 2000.

28 Ivan Damgård. Collision Free Hash Functions and Public Key Signature Schemes. In *Advances in Cryptology - EUROCRYPT'87, Workshop on the Theory and Application of of Cryptographic Techniques, Amsterdam, The Netherlands, April 13-15, 1987, Proceedings*, pages 203–216, 1987.

29 Ivan Damgård, Torben P. Pedersen, and Birgit Pfitzmann. On the existence of statistically hiding bit commitment schemes and fail-stop signatures. *J. Cryptology*, 10(3):163–194, 1997.

30 Jintai Ding and Bo-Yin Yang. Multivariates polynomials for hashing. In *Information Security and Cryptology, Third SKLOIS Conference, Inscrypt 2007, Xining, China, August 31 - September 5, 2007, Revised Selected Papers*, pages 358–371, 2007.

31 Ilya Dumer. On minimum distance decoding of linear codes. In Ed. G. Kabatianskii, editor, *Fifth Soviet-Swedish intern. workshop Information theory*, pages 50—-52, 1991.

32 Ilya Dumer, Alexey A Kovalev, and Leonid P Pryadko. Distance verification for LDPC codes. *arXiv preprint arXiv:1605.02410*, 2016.

33 Ilya Dumer, Daniele Micciancio, and Madhu Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Trans. Information Theory*, 49(1):22–37, 2003.

34 Uriel Feige, Jeong Han Kim, and Eran Ofek. Witnesses for non-satisfiability of dense random 3cnf formulas. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 497–508, 2006.

35 Matthieu Finiasz, Philippe Gaborit, and Nicolas Sendrier. Improved fast syndrome based cryptographic hash functions. In *Proceedings of ECRYPT Hash Workshop*, volume 2007, page 155, 2007.

36 Matthieu Finiasz and Nicolas Sendrier. Security bounds for the design of code-based cryptosystems. In *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, pages 88–105, 2009.

37 Robert G. Gallager. *Low-Density Parity-Check Codes*. MIT Press, 1963.

38 M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

39 Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 169–178, 2009.

40 Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 75–92, 2013.

41 Oded Goldreich. Candidate One-Way Functions Based on Expander Graphs. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation - In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*, pages 76–87. Springer, 2011.

42 Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Collision-free hashing from lattice problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(42), 1996.

43 Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.

44 Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. Essential coding theory. Draft of a Book, 2015.

45 Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil Segev. Finding collisions in interactive protocols - tight lower bounds on the round and communication complexities of statistically hiding commitments. *SIAM J. Comput.*, 44(1):193–242, 2015.

46 Iftach Haitner, Yuval Ishai, Eran Omri, and Ronen Shaltiel. Parallel hashing via list recoverability. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 173–190, 2015.

47 Shai Halevi and Silvio Micali. Practical and provably-secure commitment schemes from collision-free hashing. In *Advances in Cryptology - CRYPTO'96, 16th Annual International*

Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings, pages 201–215, 1996.

48 Masanori Hirotomo, Masami Mohri, and Masakatu Morii. A probabilistic computation method for the weight distribution of low-density parity-check codes. In *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.*, pages 2166–2170, 2005.

49 Chun-Yuan Hsiao and Leonid Reyzin. Finding collisions on a public road, or do secure hash functions need secret coins? In *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, pages 92–105, 2004.

50 Xiao-Yu Hu, Marc P. C. Fossorier, and Evangelos Eleftheriou. On the computation of the minimum distance of low-density parity-check codes. In *Proceedings of IEEE International Conference on Communications, ICC 2004, Paris, France, 20-24 June 2004*, pages 767–771, 2004.

51 Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 294–304, 2000.

52 Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography with constant computational overhead. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 433–442, 2008.

53 Ahmet B. Keha and Tolga M. Duman. Minimum distance computation of LDPC codes using a branch and cut algorithm. *IEEE Trans. Communications*, 58(4):1072–1079, 2010.

54 Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, pages 723–732, 1992.

55 Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar Signature Schemes. In *Advances in Cryptology - EUROCRYPT'99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, pages 206–222, 1999.

56 Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In *Advances in Cryptology - CRYPTO'99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, pages 19–30, 1999.

57 Simon Litsyn and Vladimir Shevelev. On ensembles of low-density parity-check codes: asymptotic distance distributions. *IEEE Transactions on Information Theory*, 48(4):887–908, 2002.

58 Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, pages 144–155, 2006.

59 Tsutomu Matsumoto and Hideki Imai. Public Quadratic Polynominal-Tuples for Efficient Signature-Verification and Message-Encryption. In *Advances in Cryptology - EUROCRYPT'88, Workshop on the Theory and Application of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings*, pages 419–453, 1988.

60 Ralph C. Merkle. A digital signature based on a conventional encryption function. In *Advances in Cryptology - CRYPTO'87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*, pages 369–378, 1987.

61 Ralph C. Merkle. A Certified Digital Signature. In *Advances in Cryptology - CRYPTO'89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 218–238, 1989.

62 Silvio Micali. Computationally sound proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000. Preliminary version in *FOCS'94*.

63 Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 33–43, 1989.

64 Jacques Patarin. Cryptoanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88. In *Advances in Cryptology - CRYPTO'95, 15th Annual International Cryptology Conference, Santa Barbara, California, USA, August 27-31, 1995, Proceedings*, pages 248–261, 1995.

65 Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, pages 145–166, 2006.

66 Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 73–85, 1989.

67 Ran Raz. Fast learning requires good memory: A time-space lower bound for parity learning. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:19, 2016.

68 Ohad Shamir. Fundamental limits of online and distributed algorithms for statistical learning and estimation. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 163–171, 2014.

69 Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *Advances in Cryptology - EUROCRYPT'98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*, pages 334–345, 1998.

70 Jacob Steinhardt, Gregory Valiant, and Stefan Wager. Memory, communication, and statistical queries. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 1490–1516, 2016.

71 Jacques Stern. A method for finding codewords of small weight. In *Coding Theory and Applications, 3rd International Colloquium, Toulon, France, November 2-4, 1988, Proceedings*, pages 106–113, 1988.

72 Alexander Vardy. The intractability of computing the minimum distance of a code. *IEEE Trans. Information Theory*, 43(6):1757–1766, 1997.

73 Christopher Wolf. Multivariate quadratic polynomials in public key cryptography. Cryptology ePrint Archive, Report 2005/393, 2005.

74 Yang Xiao and Kiseon Kim. Searching the minimum distances of LDPC codes. In *Wireless, Mobile and Multimedia Networks (ICWMMN 2008), IET 2nd International Conference on*, pages 211–214, 2008.

75 Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 80–91, 1982.

A Statistically Hiding Commitments

In this section we define a non-interactive notion of *statistically hiding commitments* (SHC), which can be viewed as a randomized version of CRH that achieves an input hiding property. We then show that positive results for CRH apply also to the case of SHC.

Similarly to a CRH, an SHC is defined by a public random string z (which can be reused for many commitments). The string z defines a mapping from an input x and secret randomness σ to a commitment string c. The commitment c should statistically hide x. It should also be *computationally binding* in the sense that given z it is infeasible to find a pair (x, σ) and (x', σ') which are consistent with the same c, where $x \neq x'$. We formalize these requirements below.

▶ **Definition 36** (Statistically Hiding Commitment)**.** A collection of functions

$$\mathcal{C} = \left\{ C_z : \{0,1\}^k \times \{0,1\}^{\rho(k)} \to \{0,1\}^{m(k)} \right\}_{z \in \{0,1\}^{s(k)}}$$

is a (non-interactive) *statistically hiding commitment* (SHC) if the following hold:

- (Efficient evaluation and sampling) There exists a pair of efficient algorithms: (a) a *commitment algorithm* C which given $(z \in \{0,1\}^s, x \in \{0,1\}^k, \sigma \in \{0,1\}^\rho)$ outputs $C_z(x, \sigma)$; and (b) a key-sampling algorithm $\mathcal{K}$ which given 1^k samples a index $z \in \{0,1\}^{s(z)}$.
- (Statistical hiding) For every pair of inputs $x, x' \in \{0,1\}^k$ we have

  $$\mathsf{SD}((z, C_z(x, \sigma)), (z, C_z(x', \sigma))) \leq \mathrm{neg}(k),$$

  where SD denotes statistical distance, z is picked by $\mathcal{K}(1^k)$, and σ is picked uniformly from $\{0,1\}^{\rho(k)}$.
- (Computational binding) For every probabilistic polynomial-time adversary Adv it holds that

  $$\Pr_{z \xleftarrow{R} \mathcal{K}(1^k)} [\mathsf{Adv}(z) = ((x, \sigma), (x', \sigma')) \text{ s.t. } x' \neq x \text{ and } C_z(x, \sigma) = C_z(x', \sigma')] \leq \mathrm{neg}(k).$$

As in the case of CRH, we consider the efficiency of SHC for any fixed public challenge z, namely z defines a function of x and s. This is justified by the fact that the same z can reused.

We now show a simple transformation of a CRH into an SHC using a randomness extractor [29, 47]. The high level idea is to apply a CRH to a secret random input α, and then mask the SHC input with randomness extracted from α. Since the CRH shrinks the input, there is residual entropy in α even when conditioned on the output of the CRH.

▶ **Theorem 37.** *Suppose* $\mathcal{H} = \left\{ h_z : \{0,1\}^k \to \{0,1\}^{m(k)} \right\}_{z \in \{0,1\}^{s(k)}}$ *is a CRH, where* $m(k) = \lfloor (1-c)k \rfloor$ *for some constant* $0 < c < 1$. *Let* $\mathsf{Ext}_k : \{0,1\}^k \times \{0,1\}^d \to \{0,1\}^{\lfloor ck/3 \rfloor}$ *be a strong* $(ck/2, \epsilon)$ *randomness extractor with error* $\epsilon(k) = \mathrm{neg}(k)$.
Then $\mathcal{C} = \left\{ C_z : \{0,1\}^{k'} \times \{0,1\}^{k+d} \to \{0,1\}^{m(k)} \right\}_{z \in \{0,1\}^{s(k)}}$, *where* $k(k')$ *is chosen such that* $\lfloor ck/3 \rfloor = k'$ *and* $C_z(x', (\alpha, \beta)) = (h_z(\alpha), \beta, x' \oplus \mathsf{Ext}_k(\alpha, \beta))$ *for* $\alpha \in \{0,1\}^k$ *and* $\beta \in \{0,1\}^d$, *is an SHC.*

Implementing Ext by a random linear function, the SHC obtained in Theorem 37 has the same algebraic degree as the underlying CRH. Moreover, if Ext is implemented by the linear-size pairwise independent hash function from [52], the SHC additionally maintains

the asymptotic circuit size of the CRH. Thus, we get a linear-size (degree-3) SHC under the assumptions of Theorem 19.

Unlike the case of CRH, there is no shrinkage requirement for SHC, hence a degree-3 SHC with locality 4 is implied by the existence of any SHC or CRH in NC^1, which is in turn implied by most standard cryptographic assumptions [1]. However, the existence of degree-2 SHC is left open. Using Theorem 37, a degree-2 SHC would be implied by the existence of a degree-2 CRH, which is one of the main questions left open by this work.

Hierarchical Functional Encryption*†

Zvika Brakerski[1], Nishanth Chandran[2], Vipul Goyal[3], Aayush Jain[4], Amit Sahai[5], and Gil Segev[6]

1 **Weizmann Institute of Science, Rehovot, IL**
zvika.brakerski@weizmann.ac.il
2 **Microsoft Research India, Bangalore, IN**
nichandr@microsoft.com
3 **CMU, Pittsburgh, USA**
goyal@cs.cmu.edu
4 **UCLA, Los Angeles, USA**
aayushjain1728@gmail.com
5 **UCLA, Los Angeles, USA**
sahai@cs.ucla.edu
6 **Hebrew University of Jerusalem, Jerusalem, IL**
segev@cs.huji.ac.il

Abstract

Functional encryption provides fine-grained access control for encrypted data, allowing each user to learn only specific functions of the encrypted data. We study the notion of *hierarchical* functional encryption, which augments functional encryption with *delegation* capabilities, offering significantly more expressive access control.

We present a *generic transformation* that converts any general-purpose public-key functional encryption scheme into a hierarchical one without relying on any additional assumptions. This significantly refines our understanding of the power of functional encryption, showing that the existence of functional encryption is equivalent to that of its hierarchical generalization.

Instantiating our transformation with the existing functional encryption schemes yields a variety of hierarchical schemes offering various trade-offs between their delegation capabilities (i.e., the depth and width of their hierarchical structures) and underlying assumptions. When starting with a scheme secure against an unbounded number of collusions, we can support *arbitrary* hierarchical structures. In addition, even when starting with schemes that are secure against a bounded number of collusions (which are known to exist under rather minimal assumptions such as the existence of public-key encryption and shallow pseudorandom generators), we can support hierarchical structures of bounded depth and width.

1998 ACM Subject Classification E.3 Data Encryption

Keywords and phrases Fuctional encryption

* In this extended abstract we present results from [23] and [25].

† Z. Brakerski is supported by the Israel Science Foundation (Grant No. 468/14), the Alon Young Faculty Fellowship, Binational Science Foundation (Grant No. 712307) and Google Faculty Research Award.
A. Sahai and A. Jain are supported in part from a DARPA/ARL SAFEWARE award, NSF Frontier Award 1413955, NSF grants 1619348, 1228984, 1136174, and 1065276, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through the ARL under Contract W911NF-15-C-0205. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.
G. Segev is supported by the European Union's 7th Framework Program (FP7) via a Marie Curie Career Integration Grant, by the Israel Science Foundation (Grant No. 483/13), by the Israeli Centers of Research Excellence (I-CORE) Program (Center No. 4/11), by the US-Israel Binational Science Foundation (Grant No. 2014632), and by a Google Faculty Research Award.

© Zvika Brakerski, Nishanth Chandran, Vipul Goyal, Aayush Jain, Amit Sahai, and Gil Segev;
licensed under Creative Commons License CC-BY

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.8

1 Introduction

The rapidly evolving vision of functional encryption [49, 15, 48] offers tremendous flexibility when accessing encrypted data. Specifically, functional encryption schemes support restricted decryption keys that allow users to learn specific functions of the encrypted data and nothing else. Motivated by the early examples of functional encryption schemes for specific functionalities (such as identity-based encryption [51, 13, 27]), extensive research has recently been devoted to the study of functional encryption (see, for example, [49, 15, 48, 38, 3, 9, 19, 30, 37, 5, 53, 32, 7, 22, 42, 21] and the references therein).

In a functional encryption scheme, a trusted authority holds a master secret key known only to the authority. When the authority is given the description of some function f as input, it uses its master secret key to generate a functional key sk_f associated with the function f. Now, anyone holding the functional key sk_f and an encryption of any message x, can compute $f(x)$ but cannot learn any additional information about the message x. Such fine-grained access to encrypted data is extremely useful for a wide variety of applications, including expressive access control, spam filtering, mining encrypted databases, and more (we refer the reader to the survey by Boneh, Sahai and Waters [16] for an in-depth discussion of these applications).

Hierarchical functional encryption. Motivated by the applicability of functional encryption to expressive access-control systems, in this paper we study the notion of *hierarchical* functional encryption, which was introduced by Ananth, Boneh, Garg, Sahai and Zhandry [4]. The hierarchical notion augments standard functional encryption with delegation capabilities, enabling significantly more expressive access control.

Specifically, recall that in a functional encryption scheme, the holder of the master secret key msk can generate a functional key sk_f corresponding to any given function f. In a hierarchical functional encryption scheme, the holder of any such functional key sk_f can in turn generate a functional key $\mathsf{sk}_{g\circ f}$ corresponding to the function $g \circ f$ for any given function g. Now, anyone holding the delegated functional key $\mathsf{sk}_{g\circ f}$ and an encryption of any message x, can compute $(g \circ f)(x) = g(f(x))$ but cannot learn any additional information about the message x. Such expressive delegation capabilities give rise to *hierarchical* access control, which is a sought-after ingredient in modern access control systems. In particular, the notion of hierarchical functional encryption generalizes those of hierarchical attribute-based encryption, hierarchical identity-based encryption and more (see the discussion at the end of Section 3 on the delegation capabilities of functional encryption).

Ananth et al. formalized a notion of security for hierarchical functional encryption schemes, and sketched how the functional encryption scheme of Garg et al. [30] can be transformed into a hierarchical one by using a general-purpose indistinguishability obfuscator [8, 30].[1] Their approach, however, is both tailored to the specific functional encryption scheme of Garg et al. [30], and can only support hierarchical structures of *constant* depth (i.e., can only support a constant number of successive delegations).[2]

[1] It was recently shown by Ananth and Jain [6] and by Bitansky and Vaikuntanathan [10] that indistinguishability obfuscation can be constructed from some flavors of functional encryption. Specifically, from *succinct* functional encryption with *sub-exponential security*. Our approach is both more direct and requires no such properties.

[2] In the hierarchical scheme of Ananth et al. [4], a delegated functional key $\mathsf{sk}_{g\circ f}$ for the function $g \circ f$ is

1.1 Our Contributions

We present a *generic transformation* that converts any general-purpose public-key functional encryption scheme into a hierarchical one without relying on any additional assumptions (and, in particular, without relying on indistinguishability obfuscation). Our transformation allows instantiations based both on unbounded-collusion functional encryption schemes and on bounded-collusion ones. This level of generality yields a variety of hierarchical schemes based on various assumptions in the standard model. These include strong assumptions such as indistinguishability obfuscation [30, 53] or multilinear maps [32], and much milder assumptions such as learning with errors [37], and even the existence of any public-key encryption scheme and low-depth pseudorandom generator [38].

We stress that our result stands even if it turns out that indistinguishability obfuscation is impossible to achieve or requires strong computational assumptions. One could view it as evidence that the hierarchical properties do not stem from the "obfuscation-like" features of functional encryption, but rather from the more rudimentary properties that are achievable under minimal assumptions.

Security and efficiency. In terms of security, the schemes resulting from our transformation guarantee semi-adaptive security. Moreover, by assuming that the underlying functional encryption scheme is sub-exponentially secure, we obtain adaptive security via a standard complexity leveraging argument. In terms of efficiency, our approach results in keys and ciphertexts with a rather low overhead compared to the efficiency of the underlying functional encryption scheme. A ciphertext in our scheme is essentially a ciphertext of the underlying scheme. A delegated functional key for $g \circ f$ contains two functional keys for functions that essentially compute f and g, respectively, in addition to some basic cryptographic computation (an evaluation of a PRF, and an encryption of a ciphertext of the underlying scheme). We refer the reader to the overview below for more details on the security and efficiency of the schemes resulting from our approach.

Instantiations. The variety of schemes resulting from our transformation offer different trade-offs between their underlying assumptions and their delegation capabilities (i.e., the depth and width of the hierarchical structures that they support). For example, instantiating our transformation with the schemes of Garg et al. [30] and Waters [53] results in hierarchical schemes that support hierarchical structures of any polynomial depth and any polynomial width (where these polynomials do not have to be specified in advance). In addition, instantiating our transformation with the schemes of Goldwasser et al. [37] or Gorbunov et al. [38] results in hierarchical schemes that support hierarchical structures of any constant depth and any pre-determined polynomial width. This should be compared to the hierarchical scheme of Ananth et al. [4] that is constructed from the specific functional encryption scheme of Garg et al. [30], and can only support hierarchical structures of *constant* depth, and to the alternative hierarchical scheme in this work (see Section 1.3) that is constructed based on even stronger assumptions and still requires an a-priori bound on the depth of the support hierarchical structures. We refer the reader to Table 1 for a comparison of the assumptions underlying the known hierarchical functional encryption schemes and of their supported hierarchical structures.

computed from sk_f by applying the obfuscator to a program that contains sk_f as part of its description. Thus, since sk_f itself consists of such an obfuscated program, this allows for only a constant number of successive delegations.

Table 1 A comparison of the assumptions underlying the known hierarchical functional encryption schemes and of their supported hierarchical structures. We note that indistinguishability obfuscation ($i\mathcal{O}$) is known to imply unbounded-collusion functional encryption [53], which in turn clearly implies bounded-collusion functional encryption. In addition, bounded-collusion functional encryption is implied by much milder assumptions such as learning with errors [37], and even the existence of any public-key encryption scheme and low-depth pseudorandom generator [38].

Assumption	**Hierarchical Structure**	
	Depth	**Width**
$i\mathcal{O}$ [4]	Constant	Unbounded
Sub-exponentially-secure $i\mathcal{O}$ and sub-exponentially-secure HIBE (this work [25])	Any fixed polynomial	Unbounded
Unbounded-collusion FE (this work [23])	Unbounded	Unbounded
Bounded-collusion FE (this work [23])	Constant	Any fixed polynomial

1.2 Overview of Our Approach

Formally, a hierarchical FE scheme contains the standard $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ algorithms, in addition to a new *delegation* algorithm $\mathsf{Delegate}$. The delegation algorithm $\mathsf{Delegate}(\mathsf{hsk}_f, g)$ is identical in syntax to the KG algorithm, except that it takes a functional key hsk_f (which can itself be the output of a previous delegation) instead of the master secret key msk. Its output is a key $\mathsf{hsk}_{g\circ f}$ corresponding to the composed function $g \circ f$.

Indeed, the way we implement this functionality is by associating a unique master secret key with any delegable functional key. Namely, a delegable key hsk_f (with respect to the master key pair $(\mathsf{msk}, \mathsf{mpk})$) will contain a fresh master secret key msk' in addition to a "standard" functional key for a re-encryption function $\mathsf{sk}_{\mathsf{ReEnc}_{f,\mathsf{mpk}'}}$ (the key pair $(\mathsf{msk}', \mathsf{mpk}')$ is generated using the standard setup procedure). The function $\mathsf{ReEnc}_{f,\mathsf{mpk}'}(x)$, intuitively, takes an input x and outputs a *functional encryption* of $f(x)$ under the new key mpk'. It is obvious that since msk' is a part of hsk_f, then the owner of $\mathsf{hsk}_f = (\mathsf{sk}_{\mathsf{ReEnc}_{f,\mathsf{mpk}'}}, \mathsf{msk}')$ can derive $f(x)$ itself if it so desires. The beauty of this procedure is that it can then be repeated. If hsk_f needs to be delegated via $\mathsf{Delegate}(\mathsf{hsk}_f, g)$, then one only needs to generate a new pair $(\mathsf{msk}'', \mathsf{mpk}'')$, use msk' to obtain $\mathsf{sk}'_{\mathsf{ReEnc}_{g,\mathsf{mpk}''}}$ and output $\mathsf{hsk}_{g\circ f} = (\mathsf{sk}_{\mathsf{ReEnc}_{f,\mathsf{mpk}'}}, \mathsf{sk}'_{\mathsf{ReEnc}_{g,\mathsf{mpk}''}}, \mathsf{msk}'')$. In the decryption process, we start with some $\mathsf{ct} = \mathsf{FE.Enc}_{\mathsf{mpk}}(x)$, use the first component of the key to obtain $\mathsf{ct}' = \mathsf{FE.Enc}_{\mathsf{mpk}'}(f(x))$, and then using the second component to obtain $\mathsf{ct}'' = \mathsf{FE.Enc}_{\mathsf{mpk}''}(g(f(x)))$. Finally, msk'' is used to decrypt ct'' and thus learn the value $g(f(x))$.

Care needs to be taken in order to securely realize the above intuition. In particular, one has to come up with a source of randomness for the re-encryption process. This is done by slightly modifying the encryption algorithm of the hierarchical scheme such that $\mathsf{Enc}(\mathsf{mpk}, x) = \mathsf{FE.Enc}(\mathsf{mpk}, (x, K, \perp))$, where K is a key to a puncturable pseudorandom function PRF, and $\perp$ is a placeholder that will only be used in the proof. Similarly, we will generate functional keys of the form $\mathsf{sk}_{\mathsf{ReEnc}_{f,t,\mathsf{mpk}',c}}$, where t is a random tag and c is a random string that will be used in the proof. The function $\mathsf{ReEnc}_{f,t,\mathsf{mpk}',c}(x, K, \perp)$ will compute $f(x)$ and encrypt the tuple $(f(x), K', \perp)$ under msk' using randomness r'. The randomness for the generation of K' and r' is produced by evaluating $\mathsf{PRF}_K(t)$.

For the sake of our security proof, one last addition is made to the description of $\mathsf{ReEnc}_{f,t,\mathsf{mpk}',c}$. If its input is of the form $(\cdot,\cdot,\mathsf{k})$, where k is a key for a symmetric encryption scheme, then the first two arguments are ignored and $\mathsf{SKE.Dec}_{\mathsf{k}}(c)$ is output. Thus we implement a "trapdoor circuit" (or a "Trojan") as per [29, 5].

Security notion. The notions of security that we consider in this work are those formalized by Ananth et al. [4]. Specifically, we consider adversaries that obtain functional keys for various functions of their choice by issuing key-generation queries and delegation queries. We require that such adversaries have only a negligible advantage in distinguishing the encryptions of two challenge messages, x_0^* and x_1^*, of their choice as long as for any function f for which they obtain a functional key it holds that $f(x_0^*) = f(x_1^*)$, where such a key may be produced either as a result of a key-generation query or a delegation query (we refer the reader to Section 3 for more details).

We prove that if the underlying scheme $\mathcal{FE}$ is selectively secure then the resulting hierarchical scheme is selectively secure, and if $\mathcal{FE}$ is semi-adaptively secure then the resulting hierarchical scheme is semi-adaptively secure.[3] We leave it as an intriguing open problem to design a hierarchical functional encryption scheme that guarantees adaptive security. We note that already in the less-expressive setting of identity-based encryption, designing adaptively-secure hierarchical schemes is extremely challenging. In particular, Lewko and Waters [46] recently showed why known proof methods fall short of proving adaptive security even for adaptively-secure hierarchical identity-based encryption (which is a special case of hierarchical FE) without degrading exponentially with the number of delegation levels.

Proof overview. Let us focus on the case of selective security, semi-adaptive security follows by a practically identical argument. In the selective security game, the adversary first specifies challenge messages x_0^* and x_1^*, receives mpk, and then makes a sequence of key-generation and delegation queries. One could visualize the structure that is generated by these queries as a tree, whose root is $(\mathsf{msk},\mathsf{mpk})$ and whose nodes are the key pairs that are generated upon each call to KG or $\mathsf{Delegate}$. Each such call generates a new child for one of the nodes in the tree, as per the adversary's choice. Each node is associated with a function f which was input to $\mathsf{KG}/\mathsf{Delegate}$ in its creation, and also with a function $\tilde{f}$, which is the composition of all functions from the root to that node. If $\tilde{f}(x_0^*) = \tilde{f}(x_1^*)$ then we say that the node is *observable*, since the adversary is allowed to see the functional key $\mathsf{hsk}_{\tilde{f}}$ associated with that node. We can assume w.l.o.g that all the leaves of the tree are observable.

The high-level intuition of the proof is the following. Let us pretend that ReEnc is actually capable of outputting a re-encrypted ciphertext which is encrypted with true randomness, rather than with pseudorandomness. Now, consider an unobservable node i (i.e., a node corresponding to f_i and $\tilde{f}_i$ for which $\tilde{f}_i(x_0^*) \neq \tilde{f}_i(x_1^*)$) that only has observable children. This means that all functions $\mathsf{ReEnc}_{f,t,\mathsf{mpk}',c}$ that are generated relative to this node's msk_i will output the same value whether the challenge ciphertext is an encryption of x_0^* or of x_1^*. The security of the underlying scheme will guarantee that the re-encrypted ciphertext cannot be used to distinguish x_0^* from x_1^*. Let us take another leap of faith and pretend that

[3] We briefly remind the reader the differences between selective, semi-adaptive, and adaptive security. *Selective* security considers adversaries that specify their challenge messages before seeing the public parameters or making any key queries. *Semi-adaptive* security, as recently defined by Chen and Wee [26], considers adversaries that specify their challenge messages after seeing the public parameters but before making any key queries. Finally, *adaptive* security considers adversaries that specify their challenge messages even after seeing the public parameters and making key queries.

not only the re-encrypted ciphertext cannot distinguish x_0^* from x_1^* but it is in fact identical in both cases. Then the above process can propagate towards the root of the tree, where at every step we increase the number of nodes whose output is the same regardless of whether the challenge ciphertext encrypts x_0^* or x_1^*. Once this process gets all the way to the root and applied to the challenge ciphertext itself, the proof is complete.

This intuition is implemented using the mechanisms of punctured programming [50] and "trapdoor circuits" [29] (or "Trojans" [5]). We will replace the c values in $\mathsf{ReEnc}_{f,t,\mathsf{mpk}',c}$ with symmetric encryptions of our "fantasy ciphertexts" (ones that are encrypted with true randomness), and append the challenge ciphertext with the appropriate symmetric decryption key (in fact, multiple symmetric keys will be needed, one for every level of the hierarchy, and one has to carefully control their propagation along the tree). Puncturable PRFs will be used to argue that the use of fantasy ciphertexts is indistinguishable from the actual output of $\mathsf{ReEnc}_{f,t,\mathsf{mpk}',c}$, which will allow the proof idea from above to go through. This requires a careful and delicate argument since we can only puncture a PRF key that had been generated with fresh randomness, hence one has to also consider fantasy PRF keys and propagate them along the tree as well together with the fantasy ciphertexts. The formal proof thus contains many fine points and a large number of steps, and is provided in Section 4.

1.3 The Multi-Authority Setting and an Alternative Hierarchical Scheme

While above, we have focused on the question of delegating functional keys to other parties, a closely related problem is that of achieving functional encryption in the context of multiple key-issuing authorities. In Multi-Authority Functional Encryption (MAFE), we allow n authorities to "independently" generate their private and public keys. An encryptor should be able to encrypt a message m along with a policy F over the various authorities. Any authority i, should be able to generate a token for a user with identity UID and property U_i. A user with identity UID with tokens for U_i from authority $i \in [n]$, should be able to decrypt the ciphertext to recover $F(U_1, .., U_n, m)$. We require that colluding users, say UID_1 and UID_2, (possibly, in collusion with some corrupt authorities) should not jointly learn anything more from the ciphertext than what they are authorized to.

Our main construction idea is natural – we view a ciphertext as an obfuscated program and functional keys as signatures generated with respect to each authority's unique public key. To elucidate this main idea, let us first analyze the problem in a single authority scenario. The starting point of our construction is a construction of a functional encryption scheme due to [19] based on *differing-inputs obfuscation* (diO). In their construction, a ciphertext is an obfuscated circuit that checks the signature for a function f and computes f if the signature is valid. The security proof relied on the fact that if the adversary distinguishes the ciphertext (or rather an obfuscated circuit) encrypting m_0 from that of m_1, then one must be able to extract a signature on a function g such that $g(m_0) \neq g(m_1)$ thereby producing a forgery. This construction is easily scalable to the multi-authority setting. However, given the implausibility of differing-inputs obfuscation [31], our objective is a construction based on indistinguishability obfuscation (iO). While it was observed in [19] that iO behaves like diO for the setting where the circuits in question differ only on a polynomially bounded number of points, unfortunately, there are simply too many points on which these functions would differ for this strategy to yield a construction from iO.

Constructing MAFE based on indistinguishability obfuscation. Our first idea is to use signatures that are *unique*, in the sense, that for every message there exists only one signature that verifies. In order to instantiate unique signatures, we use puncturable PRF's, indistinguishability obfuscation and an injective one way function following the punctured programming technique from [50]. The main idea of the security proof is that we can build exponentially many hybrids corresponding to the function space. We index each hybrid with a function x. In hybrid x, the ciphertext takes as input f with a signature σ and checks if $f < x$. If that is the case, it outputs $f(m_0)$ otherwise $f(m_1)$. We argue indistinguishability between hybrids x and $x+1$ by intermediate hybrids where we puncture the PRF at x and replace the PRF evaluations with a random sample. Then we further note that if the adversary now distinguishes these hybrids, then it can be used to invert an injective one way function in sub-exponential time. In order to handle key corruptions, we now let the signing key be an obfuscation of a circuit that evaluates a PRF on the input. This allows us to "program" the secret keys using puncturing techniques [50] and the techniques described above so that the proof can be made to go through.

Constructing MAFE based on LWE. Since MAFE for arbitrary policies imply obfuscation it is unlikely to construct them from simpler assumptions. We also study a very natural policy referred to as the $n - out - of - n$ threshold policy. In this policy for any function f the decryptor must get tokens from all the authorities to learn $f(m)$. We describe our construction at a very high level below. Our starting point is the recent construction of threshold homomorphic encryption (TFHE) by [47]. In a TFHE scheme the secret key can be shared among n parties and any (evaluated) ciphertext can be partially decrypted using each of the n key shares. The partial decryptions can be finally added to yield the output. The security guarantee is two fold: First, given at most $n-1$ key shares the semantic security of the encryption holds. Second, partial decryption of any ciphertext using any key share can be statistically simulated using the remaining $n-1$ shares and the output of the decryption. The (simplified) scheme can be described as follows. Each authority runs an FE setup. To encrypt a message m, the encryptor runs a fresh setup of a TFHE system. He encrypts m as ct and outputs n FE ciphertexts, one corresponding to each authority. For authority i he encrypts $(\mathsf{ct}, \mathsf{sk}_i, *)$ where $*$ represents some additional strings (for example, PRF keys, e.t.c used for programming the proof) and sk_i is i^{th} partial TFHE decryption key. The key for a function f is now simply an FE key for a function that takes as input a TFHE ciphertext ct and evaluates it for function f and then computes a partial decryption using sk_i.

The solution described above suffers from one key flaw. The problem is that the scheme is not *user-collusion-resistant*. One user uid_1 can query tokens for f from say k authorities, while another user uid_2 can query them from other $n-k$ authorities and later collude to recover $f(m)$. We prevent this by relying on Pseudo-Random Zero-Sharing (PRSS) [28]. Using this primitive one can secret share 0 pseudorandomly and deterministically using initially distributed shares $(\beta_1, .., \beta_n)$ and any common input. Concretely there exists a publicly known function g such that for any index x, $\{g(\beta_i, x)\}_{i\in[n]}$ forms a pseudorandom secret sharing of 0. We now modify the existing scheme as follows: FE ciphertext for each authority i now also consists β_i and the FE key now produces TFHE partial decryptions masked by $g(\beta_i, uid, f)$. This masking prevents the adversary to collude in the manner described above.

Applications. We also show that MAFE immediately implies multi-authority attribute based encryption and this yields the first decentralised ABE scheme without setup and without the use of random oracles.

Alternate construction of Hierarchical FE. An application of the unique signature idea described above is that it can be extended to allow for delegation of function keys. In what we described above, if the signature scheme is delegatable (i.e. if it allows us to compute a signature on $f||g$ for any g given a signature on f) then it allows us to compute keys for $g \cdot f$ given a key for f, where $g \cdot f$ denotes the composition of functions f and g and on input x computes $g(f(x))$. In order to construct this primitive, we make use of the exponential hybrid approach and make interesting use of hierarchical identity based encryption (HIBE). The main idea behind this construction is as follows: to encrypt a message m, we let the ciphertext be an obfuscated program that takes as input a "function" f. On input f, the obfuscated program will compute a HIBE encryption of $f(m)$ on identity f using randomness generated from a PRF applied on f. Any decryptor can then decrypt this ciphertext using a HIBE key for identity f or its prefix. Here too, the security proof goes input by input over the space of all circuits. Our construction supports delegation up to any (a-priori bounded) polynomial number of times, in contrast to prior schemes which allowed delegation only up to $O(1)$ number of times [4].

A detailed description of the MAFE construction as well as the alternate construction for HFE can be found in [25].

1.4 Related Work

Hierarchical encryption schemes. Encryption schemes supporting a hierarchical structure have been extensively studied in the setting of identity-based encryption, and have been recently studied in the more general setting of attribute-based encryption and functional encryption.

The line of research on hierarchical identity-based encryption has been extremely successful, starting with schemes in the random oracle model, evolving through selectively-secure schemes in the standard model and graduating to adaptively secure schemes for polynomially many levels. It is far beyond the scope of this paper to provide an extensive overview of this line of research, and we refer the reader to [34, 40, 11, 12, 18, 33, 52, 1, 2, 43, 44, 45, 24, 46] and the references therein.

Recently, Boneh et al. [14] constructed an attribute-based encryption scheme that supports delegation of keys. This scheme enables anyone holding a key sk_P corresponding to a predicate P to generate a key $\mathsf{sk}_{P \wedge Q}$ corresponding to the predicate $P \wedge Q$ for any given predicate Q. Now, given the key $\mathsf{sk}_{P \wedge Q}$ and an encryption of any pair (x, m), one can recover m if and only if $(P \wedge Q)(x) = 1$. Although the setting of attribute-based encryption is significantly more expressive than the identity-base one, it does not seem to come close to the general setting of functional encryption that we consider in this paper.

Finally, as discussed above, Ananth et al. [4] sketched how the functional encryption scheme of Garg et al. [30] can be transformed into a hierarchical one by using a general-purpose indistinguishability obfuscator. When compared to their scheme our approach offers two main advantages. First, whereas Ananth et al. rely on a specific scheme and on indistinguishability obfuscation, we can rely on any underlying general-purpose scheme. This enables us to rely on a variety of underlying assumptions, including learning with errors and even the existence of any public-key encryption scheme and low-depth pseudorandom

generators, as discussed in Section 1.1. Furthermore, as new functional encryption schemes are presented, they can immediately be plugged in to our construction. Second, the schemes resulting from our transformation guarantee semi-adaptive security, whereas the scheme of Ananth et al. guarantees only the somewhat less realistic notion of selective security.

Encapsulation techniques in functional encryption. Key encapsulation is a very useful approach for improving both the efficiency and the security of encryption schemes. Specifically, key encapsulation typically means that instead of encrypting a message x under a fixed key sk, one can instead sample a fresh key k, encrypt x under k, and then encrypt k under sk. Recently, Ananth et al. [5], followed by Brakerski et al. [21], showed that key encapsulation is useful also for functional encryption, and can be used for generically enhancing the functionality and the security of such schemes. Their approaches suggest that encapsulation techniques may in fact be a general tool that is useful in the design of functional encryption schemes. As discussed in Section 1.2, our construction relies on encapsulation techniques as a key ingredient, significantly extending the initial ideas of Ananth et al. and Brakerski et al. from encapsulating keys to realizing a re-encryption mechanism that generates a hierarchical structure.

1.5 Paper Organization

The remainder of this paper is organized as follows. In Section 2 we provide an overview of the notation, definitions, and tools underlying our constructions. In Section 3 we present the notion of a hierarchical functional encryption scheme and define its security. In Section 4 we present our generic construction of a hierarchical functional encryption scheme.

The details of the results described in Section 1.3 are deferred from this extended abstract and can be found in [25].

2 Preliminaries

In this section we present the notation and basic definitions that are used in this work. For a distribution X we denote by $x \leftarrow X$ the process of sampling a value x from the distribution X. Similarly, for a set $\mathcal{X}$ we denote by $x \leftarrow \mathcal{X}$ the process of sampling a value x from the uniform distribution over $\mathcal{X}$. For an integer $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$. Throughout the paper, we denote by λ the security parameter. A function $\mathsf{neg} : \mathbb{N} \to \mathbb{R}$ is *negligible* if for every constant $c > 0$ there exists an integer N_c such that $\mathsf{neg}(\lambda) < \lambda^{-c}$ for all $\lambda > N_c$. Two sequences of random variables $X = \{X_\lambda\}_{\lambda \in \mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda \in \mathbb{N}}$ are *computationally indistinguishable* if for any probabilistic polynomial-time algorithm $\mathcal{A}$ there exists a negligible function $\mathsf{neg}(\cdot)$ such that $\left|\Pr[\mathcal{A}(1^\lambda, X_\lambda) = 1] - \Pr[\mathcal{A}(1^\lambda, Y_\lambda) = 1]\right| \leq \mathsf{neg}(\lambda)$ for all sufficiently large $\lambda \in \mathbb{N}$.

2.1 Pseudorandom Functions

Let $\{\mathcal{K}_\lambda, \mathcal{X}_\lambda, \mathcal{Y}_\lambda\}_{\lambda \in \mathbb{N}}$ be a sequence of sets and let $\mathcal{PRF} = (\mathsf{PRF.Gen}, \mathsf{PRF.Eval})$ be a function family with the following syntax:

- $\mathsf{PRF.Gen}$ is a probabilistic polynomial-time algorithm that takes as input the unary representation of the security parameter λ, and outputs a key $K \in \mathcal{K}_\lambda$.
- $\mathsf{PRF.Eval}$ is a deterministic polynomial-time algorithm that takes as input a key $K \in \mathcal{K}_\lambda$ and a value $x \in \mathcal{X}_\lambda$, and outputs a value $y \in \mathcal{Y}_\lambda$.

The sets $\mathcal{K}_\lambda$, $\mathcal{X}_\lambda$, and $\mathcal{Y}_\lambda$ are referred to as the *key space*, *domain*, and *range* of the function family, respectively. For $K \in \mathcal{K}_\lambda$, we use the notation $\mathsf{PRF.Eval}(K,\cdot)$, $\mathsf{PRF.Eval}_K(\cdot)$ and $\mathcal{PRF}_K(\cdot)$ interchangeably.

The following is the standard definition of a pseudorandom function family.

▶ **Definition 2.1** (Pseudorandomness)**.** A function family $\mathcal{PRF} = (\mathsf{PRF.Gen}, \mathsf{PRF.Eval})$ is *pseudorandom* if for every probabilistic polynomial-time algorithm $\mathcal{A}$ there exists a negligible function $\mathsf{neg}(\cdot)$ such that

$$\begin{aligned}\mathsf{Adv}_{\mathcal{PRF},\mathcal{A}}(\lambda) &\stackrel{\mathsf{def}}{=} \left| \Pr_{K\leftarrow\mathsf{PRF.Gen}(1^\lambda)}\left[\mathcal{A}^{\mathsf{PRF.Eval}_K(\cdot)}(1^\lambda)=1\right] - \Pr_{f\leftarrow F_\lambda}\left[\mathcal{A}^{f(\cdot)}(1^\lambda)=1\right]\right| \\ &\leq \mathsf{neg}(\lambda),\end{aligned}$$

for all sufficiently large $\lambda \in \mathbb{N}$, where F_λ is the set of all functions that map $\mathcal{X}_\lambda$ into $\mathcal{Y}_\lambda$.

In addition to the standard notion of a pseudorandom function family, we rely on the seemingly stronger (yet existentially equivalent) notion of a *puncturable* pseudorandom function family [41, 17, 50, 20]. In terms of syntax, this notion asks for an additional probabilistic polynomial-time algorithm, $\mathsf{PRF.Punc}$, that takes as input a key $K \in \mathcal{K}_\lambda$ and a set $S \subseteq \mathcal{X}_\lambda$ and outputs a "punctured" key K_S. The properties required by such a puncturing algorithm are captured by the following definition.

▶ **Definition 2.2** (Puncturable PRF)**.** A pseudorandom function family $\mathcal{PRF} = (\mathsf{PRF.Gen}, \mathsf{PRF.Eval}, \mathsf{PRF.Punc})$ is *puncturable* if the following properties are satisfied:

1. **Functionality:** For all sufficiently large $\lambda \in \mathbb{N}$, for every set $S \subseteq \mathcal{X}_\lambda$, and for every $x \in \mathcal{X}_\lambda \setminus S$ it holds that

$$\Pr_{\substack{K\leftarrow\mathsf{PRF.Gen}(1^\lambda);\\ K_S\leftarrow\mathsf{PRF.Punc}(K,S)}}[\mathsf{PRF.Eval}_K(x) = \mathsf{PRF.Eval}_{K_S}(x)] = 1.$$

2. **Pseudorandomness at punctured points:** Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be any probabilistic polynomial-time algorithm such that $\mathcal{A}_1(1^\lambda)$ outputs a set $S \subseteq \mathcal{X}_\lambda$, a value $x \in S$, and state information state. Then, for any such $\mathcal{A}$ there exists a negligible function $\mathsf{neg}(\cdot)$ such that

$$\begin{aligned}\mathsf{Adv}_{\mathcal{PRF},\mathcal{A}}(\lambda) &\stackrel{\mathsf{def}}{=} \left|\Pr\left[\mathcal{A}_2(K_S, \mathsf{PRF.Eval}_K(x), \mathsf{state})=1\right] - \Pr\left[\mathcal{A}_2(K_S, y, \mathsf{state})=1\right]\right| \\ &\leq \mathsf{neg}(\lambda)\end{aligned}$$

for all sufficiently large $\lambda \in \mathbb{N}$, where $K \leftarrow \mathsf{PRF.Gen}(1^\lambda)$, $(S, x, \mathsf{state}) \leftarrow \mathcal{A}_1(1^\lambda)$, $K_S = \mathsf{PRF.Punc}(K, S)$, and $y \leftarrow \mathcal{Y}_\lambda$.

For our constructions we rely on pseudorandom functions that need to be punctured only at a single point (i.e., in both parts of Definition 2.2 it holds that $S = \{x^*\}$ for some $x^* \in \mathcal{X}_\lambda$). As observed by [41, 17, 50, 20] the GGM construction [36] of PRFs from any one-way function can be easily altered to yield such a puncturable pseudorandom function family.

Augmented evaluation. When dealing with pseudorandom functions that need to be punctured only at a single point, we find it natural to consider an "augmented" evaluation algorithm that outputs a pre-determined value y^* at the punctured point. That is, we extend the functionality of $\mathsf{PRF.Eval}$ such that given an augmented key of the form $(K_{x^*}, (x^*, y^*))$, it holds that

$$\mathsf{PRF.Eval}_{(K_{x^*},(x^*,y^*))}(x) = \begin{cases} y^*, & \text{if } x = x^* \\ \mathsf{PRF.Eval}_{K_{x^*}}(x), & \text{if } x \neq x^* \end{cases}.$$

2.2 Private-Key Encryption with Pseudorandom Ciphertexts

A private-key encryption scheme over a message space $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ is a triplet $\Pi = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ of probabilistic polynomial-time algorithms. The key-generation algorithm KG takes as input the unary representation 1^λ of the security parameter $\lambda \in \mathbb{N}$ and outputs a secret key k. The encryption algorithm Enc takes as input a secret key k and a message $x \in \mathcal{X}_\lambda$, and outputs a ciphertext c. The decryption algorithm Dec takes as input a secret key k and a ciphertext c, and outputs a message $x \in \mathcal{X}_\lambda$ or the dedicated symbol $\perp$. In terms of correctness we require that for any key k that is produced by $\mathsf{KG}(1^\lambda)$ and for every message $x \in \mathcal{X}_\lambda$ it holds that $\mathsf{Dec}(\mathsf{k}, \mathsf{Enc}(\mathsf{k}, x)) = x$ with probability 1 over the internal randomness of the algorithms Enc and Dec. We also require that a uniformly distributed string does not decrypt to a valid message with overwhelming probability, i.e. $\mathsf{Dec}(\mathsf{k}, \rho) = \perp$ with probability $(1 - \mathsf{neg}(\lambda))$ over the randomness of the key k and a uniformly distributed string ρ of the same length as the ciphertext[4]. In terms of security, we rely on the following standard notion of pseudorandom ciphertexts which can be based on any one-way function (see, for example, [35]).

▶ **Definition 2.3** (Pseudorandom ciphertexts)**.** A private-key encryption scheme $\Pi = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ has *pseudorandom ciphertexts* if for any probabilistic polynomial-time adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, there exists a negligible function $\mathsf{neg}(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{PC}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Exp}^{\mathsf{PC}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2}\right| \leq \mathsf{neg}(\lambda),$$

for all sufficiently large $\lambda \in \mathbb{N}$, where the random variable $\mathsf{Exp}^{\mathsf{PC}}_{\Pi,\mathcal{A}}(\lambda)$ is defined via the following experiment:

1. $\mathsf{k} \leftarrow \mathsf{KG}(1^\lambda)$, $b \leftarrow \{0,1\}$.
2. $(x^*, \mathsf{state}) \leftarrow \mathcal{A}_1^{\mathsf{Enc}(\mathsf{k},\cdot)}(1^\lambda)$, where $x^* \in \mathcal{X}_\lambda$.
3. $c_0^* \leftarrow \mathsf{Enc}(\mathsf{k}, x^*)$, $c_1^* \leftarrow \{0,1\}^{|c_0^*|}$.
4. $b' \leftarrow \mathcal{A}_2^{\mathsf{Enc}(\mathsf{k},\cdot)}(c_b^*, \mathsf{state})$.
5. If $b' = b$ then output 1, and otherwise output 0.

2.3 Public-Key Functional Encryption

A public-key functional encryption scheme over a message space $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ and a function space $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ is a quadruple $\Pi = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ of probabilistic polynomial-time algorithms. The setup algorithm Setup takes as input the unary representation 1^λ of the security parameter $\lambda \in \mathbb{N}$ and outputs a master-secret key msk and a master-public key mpk. The key-generation algorithm KG takes as input a master-secret key msk and a function $f \in \mathcal{F}_\lambda$, and outputs a functional key sk_f. The encryption algorithm Enc takes as input a master-public key mpk and a message $x \in \mathcal{X}_\lambda$, and outputs a ciphertext ct. In terms of correctness we require that for all sufficiently large $\lambda \in \mathbb{N}$, for every function $f \in \mathcal{F}_\lambda$ and message $x \in \mathcal{X}_\lambda$ it holds that $\mathsf{Dec}(\mathsf{KG}(\mathsf{msk}, f), \mathsf{Enc}(\mathsf{mpk}, x)) = f(x)$ with all but a negligible probability over the internal randomness of the algorithms Setup, KG, and Enc.

We rely on the standard indistinguishability-based notion of adaptive security for public-key functional encryption (see, for example, [15, 48, 9, 5]), asking that encryptions of any

[4] More accurately, since the ciphertext length may (in general) not be a fixed function of the security parameter, the uniform string ρ is sampled as follows: Given the key k, encrypt a fixed message (say, the message 0) to obtain a ciphertext c, and then uniformly sample $\rho \leftarrow \{0,1\}^{|c|}$, where $|c|$ denotes the bit-length of c.

two messages, x_0 and x_1, are computationally indistinguishable given access to functional keys for any function f such that $f(x_0) = f(x_1)$.

▶ **Definition 2.4** (Adaptive security)**.** A functional encryption scheme $\Pi = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ over a message space $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ and a function space $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ is *adaptively secure* if for any probabilistic polynomial-time adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ there exists a negligible function $\mathsf{neg}(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{FE}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Exp}^{\mathsf{FE}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2}\right| \leq \mathsf{neg}(\lambda),$$

for all sufficiently large $\lambda \in \mathbb{N}$, where the random variable $\mathsf{Exp}^{\mathsf{FE}}_{\Pi,\mathcal{A}}(\lambda)$ is defined via the following experiment:

1. $(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{Setup}(1^\lambda)$, $b \leftarrow \{0,1\}$.
2. $(x_0^*, x_1^*, \mathsf{state}) \leftarrow \mathcal{A}_1^{\mathsf{KG}(\mathsf{msk},\cdot)}(1^\lambda, \mathsf{mpk})$, where $x_0^*, x_1^* \in \mathcal{X}_\lambda$, and for each function $f \in \mathcal{F}_\lambda$ with which $\mathcal{A}_1$ queries $\mathsf{KG}(\mathsf{msk}, \cdot)$ it holds that $f(x_0^*) = f(x_1^*)$.
3. $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, x_b^*)$.
4. $b' \leftarrow \mathcal{A}_2^{\mathsf{KG}(\mathsf{msk},\cdot)}(\mathsf{ct}^*, \mathsf{state})$, where for each function $f \in \mathcal{F}_\lambda$ with which $\mathcal{A}_2$ queries $\mathsf{KG}(\mathsf{msk}, \cdot)$ it holds that $f(x_0^*) = f(x_1^*)$.
5. If $b' = b$ then output 1, and otherwise output 0.

In addition to the above notion of adaptive security we consider two natural relaxations: semi-adaptive security, and selective security. Semi-adaptive security is defined via an experiment $\mathsf{Exp}^{\mathsf{semiFE}}_{\Pi,\mathcal{A}}(\lambda)$ that is obtained from the experiment $\mathsf{Exp}^{\mathsf{FE}}_{\Pi,\mathcal{A}}(\lambda)$ by asking the adversary to determine the challenge messages before making any key-generation queries (but *after* receiving the public key). Selective security is defined via an experiment $\mathsf{Exp}^{\mathsf{selFE}}_{\Pi,\mathcal{A}}(\lambda)$ that is obtained from the experiment $\mathsf{Exp}^{\mathsf{FE}}_{\Pi,\mathcal{A}}(\lambda)$ by asking the adversary to determine the challenge messages in advance (i.e., *before* receiving the public key).

Known constructions. General-purpose functional encryption schemes that satisfy the above notion of adaptive security are known to exist based on a variety of assumptions. Ananth et al. [5] gave a generic transformation from selective security to adaptive security, implying that schemes that are adaptively secure for any number of key-generation queries can be based on indistinguishability obfuscation [30, 53], differing-input obfuscation [19, 4], and multilinear maps [32]. In addition, schemes that are adaptively secure for a bounded number $B = B(\lambda)$ of key-generation queries can be based on the Learning with Errors (LWE) assumption (where the length of ciphertexts grows with B and with a bound on the depth of allowed functions) [37], based on any public-key encryption scheme and pseudorandom generators computable by small-depth circuits (where the length of ciphertexts grows with B and with an upper bound on the circuit size of the functions) [38], and even based on any public-key encryption scheme (for $B = 1$) [38].

3 Hierarchical Functional Encryption

In this section we define the notion of a hierarchical functional encryption scheme and formalize several notions of security for such schemes (based on [4]). A hierarchical functional encryption scheme is a functional encryption scheme that supports delegation of functional keys: Given a functional key sk_f corresponding to a function f, and given a function g, it is possible to efficiently compute a functional key $\mathsf{sk}_{g\circ f}$ corresponding to the function $g \circ f$

(i.e., the function that first applies f and then applies g). This capability is provided via a delegation algorithm denote $\mathsf{Delegate}$.

Formally, a hierarchical functional encryption scheme over a message space $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ and a function space $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ is a tuple $\Pi = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Delegate})$ of probabilistic polynomial-time algorithms, where $(\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ is a functional encryption scheme (see Section 2.3), and $\mathsf{Delegate}$ is a delegation algorithm that operates as follows: It takes as input a functional key sk_f (which had been produced either by the key-generation algorithm or by the delegation algorithm itself) corresponding to a function $f \in \mathcal{F}_\lambda$, and a function $g \in \mathcal{F}_\lambda$, and outputs a functional key $\mathsf{sk}_{g\circ f}$.

Correctness. In terms of correctness we require that for every $\lambda \in \mathbb{N}$, for every polynomial $\ell = \ell(\lambda)$, for every sequence of functions $f_1, \ldots, f_\ell \in \mathcal{F}_\lambda$, and for every message $x \in \mathcal{X}_\lambda$, it holds that

$$\mathsf{Dec}(\mathsf{sk}_{f_\ell\circ\cdots\circ f_1}, \mathsf{Enc}(\mathsf{mpk}, x)) = (f_\ell \circ \cdots \circ f_1)(x)$$

with all but a negligible probability over the internal randomness of the algorithms Setup, KG, Enc and $\mathsf{Delegate}$, where $\mathsf{sk}_{f_1} \leftarrow \mathsf{KG}(\mathsf{msk}, f_1)$ and $\mathsf{sk}_{f_{i+1}\circ\cdots\circ f_1} \leftarrow \mathsf{Delegate}(\mathsf{sk}_{f_i\circ\cdots\circ f_1}, f_{i+1})$ for every $i \in [\ell - 1]$. One can also consider schemes that support ℓ delegation levels for some *fixed* polynomial $\ell = \ell(\lambda)$, although we note that our scheme in this paper supports *any* polynomial number of delegation levels.

Security. As in the work of Ananth et al. [4, Appendix E] we consider the natural extensions of the existing indistinguishability-based definitions of functional encryption [15, 48] to the hierarchical setting. Specifically, we consider adversaries that obtain functional keys for various functions of their choice by issuing key-generation queries and delegation queries. We require that such adversaries have only a negligible advantage in distinguishing the encryptions of two challenge messages, x_0^* and x_1^*, of their choice as long as for any function f for which they obtain a functional key it holds that $f(x_0^*) = f(x_1^*)$.

The experiment $\mathsf{Exp}^{\mathsf{HFE}}_{\Pi,\mathcal{A}}(\lambda)$. Let $\Pi = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Delegate})$ be a hierarchical public-key functional encryption scheme over a message space $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ and a function space $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$, and let $\mathcal{A}$ be a probabilistic polynomial-time adversary. For each $\lambda \in \mathbb{N}$ we denote by $\mathsf{Exp}^{\mathsf{HFE}}_{\Pi,\mathcal{A}}(\lambda)$ the random variable that is defined via the following experiment involving the scheme Π, the adversary $\mathcal{A}$, and a challenger:

1. **Setup phase:** The challenger samples $(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and $b \leftarrow \{0, 1\}$.
2. **Pre-challenge phase:** $\mathcal{A}$ on input $(1^\lambda, \mathsf{mpk})$ adaptively issues queries of the form $(f, \mathsf{parent}, \mathsf{mode})$, where $f \in \mathcal{F}_\lambda$, $\mathsf{parent} \in \mathbb{N} \cup \{0\}$ and $\mathsf{mode} \in \{\mathsf{OutputKey}, \mathsf{StoreKey}\}$. The ith query $(f_i, \mathsf{parent}_i, \mathsf{mode}_i)$ is answered by the challenger as follows:
   - **a.** If $\mathsf{parent} = 0$ then the challenger generates $\mathsf{hsk}_i \leftarrow \mathsf{KG}(\mathsf{msk}, f)$.
   - **b.** Else, if $\mathsf{hsk}_{\mathsf{parent}_i}$ had already been generated (and is not $\perp$), then the challenger generates $\mathsf{hsk}_i \leftarrow \mathsf{Delegate}(\mathsf{hsk}_{\mathsf{parent}_i}, f)$. Otherwise set $\mathsf{hsk}_i = \perp$.
   - **c.** Finally, if $\mathsf{mode}_i = \mathsf{OutputKey}$ then the challenger outputs hsk_i, and if $\mathsf{mode} = \mathsf{StoreKey}$ then the challenger outputs $\perp$.
3. **Challenge phase:** $\mathcal{A}$ outputs $(x_0^*, x_1^*) \in \mathcal{X}_\lambda \times \mathcal{X}_\lambda$, and then the challenger computes $\mathsf{ct}^* \leftarrow \mathsf{Enc}(\mathsf{mpk}, x_b^*)$ and sends it to $\mathcal{A}$.
4. **Post-challenge phase:** $\mathcal{A}$ adaptively issues queries as in the pre-challenge phase.
5. **Output phase:** $\mathcal{A}$ outputs b', and the output of the experiment is 1 if and only if $b' = b$.

Valid adversaries. As standard in functional encryption, we rule out adversaries that can easily distinguish between the two challenge messages, x_0^* and x_1^*, using their queries. Specifically, we say that an adversary is *valid* if for any query $(f_i, \mathsf{parent}_i, \mathsf{mode}_i)$ where $\mathsf{mode}_i = \mathsf{OutputKey}$, it holds that $\tilde{f}_i(x_0^*) = \tilde{f}_i(x_1^*)$, where $\tilde{f}$ is defined recursively by $\tilde{f}_i = f_i \circ \tilde{f}_{\mathsf{parent}_i}$ and $f_0(x) = x$ (if any of these values is not well defined then $\tilde{f}_i(x) \equiv \perp$ for all x).

Having defined the experiment $\mathsf{Exp}^{\mathsf{HFE}}_{\Pi,\mathcal{A}}(\lambda)$ and the notion of a valid adversary, we are now ready to present our notion of adaptive security for hierarchical functional encryption schemes.

▶ **Definition 3.1.** A hierarchical functional encryption scheme $\Pi = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Delegate})$ over a message space $\mathcal{X} = \{\mathcal{X}_\lambda\}_{\lambda\in\mathbb{N}}$ and a function space $\mathcal{F} = \{\mathcal{F}_\lambda\}_{\lambda\in\mathbb{N}}$ is *adaptively secure* if for any probabilistic polynomial-time valid adversary $\mathcal{A}$ there exists a negligible function $\mathsf{neg}(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{HFE}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Exp}^{\mathsf{HFE}}_{\Pi,\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2}\right| \leq \mathsf{neg}(\lambda),$$

for all sufficiently large $\lambda \in \mathbb{N}$.

In addition to our notion of adaptive security we consider two natural relaxations: semi-adaptive security, and selective security. Semi-adaptive security is defined via an experiment $\mathsf{Exp}^{\mathsf{semiHFE}}_{\Pi,\mathcal{A}}(\lambda)$ that is obtained from the experiment $\mathsf{Exp}^{\mathsf{HFE}}_{\Pi,\mathcal{A}}(\lambda)$ by eliminating the pre-challenge query phase (note that the adversary determines the challenge messages *after* receiving the public key). Selective security is defined via an experiment $\mathsf{Exp}^{\mathsf{selHFE}}_{\Pi,\mathcal{A}}(\lambda)$ that is obtained from the experiment $\mathsf{Exp}^{\mathsf{HFE}}_{\Pi,\mathcal{A}}(\lambda)$ by asking the adversary to determine the challenge messages in advance (i.e., *before* receiving the public key).

Discussion: The delegation capabilities of functional encryption. It is important to point out that given a functional key sk_f, one cannot hope to delegate anything beyond the set of functions $g \circ f$ while maintaining the security properties of functional encryption. To see this, assume towards contradiction that there exists a function h such that h cannot be expressed as $g \circ f$, but sk_h can be derived from sk_f. Since the value of $h(x)$ cannot be inferred just by examining the value of $f(x)$, there must exist two inputs, x_0 and x_1 such that $f(x_0) = f(x_1)$ but $h(x_0) \neq h(x_1)$. Given sk_f, therefore, one should not be able to distinguish encryptions of x_0 and x_1, but by delegating to sk_h, this becomes possible, hence the contradiction.

The above optimality claim may seem a little confusing when we think about special cases such as attribute-based encryption (ABE) or even identity-based encryption (IBE). In ABE for example, each ciphertext contains an attribute x and a message m, and sk_f reveals m if and only if $f(x) = 1$. In hierarchical ABE (HABE) [39, 14], given sk_f, one should be able to derive $\mathsf{sk}_{f\wedge f'}$ for all f'. At first glance, this seems to not be covered by our definition since $f \wedge f'$ cannot be expressed as $g \circ f$. However, we notice that in fact when thinking about HABE as a special case of functional encryption, it must be the case that what we call sk_f, is in fact a functional key for the function $f^+(x, m) = ((f(x) = 1)?(x, m) : \perp)$ (i.e., the function that takes (x, m) as input, and if $f(x) = 1$ it returns (x, m) and otherwise it returns $\perp$). This is because if $f(x) = 1$ then x can always be recovered by considering delegated keys that fix the value of each bit of x to 0 or 1, and check if decryption still works. It is clear from this viewpoint that $(f \wedge f')^+$ can be seen as $g \circ f^+$ for an appropriate g. Therefore, our definition and construction are fully compatible also with the more restricted settings of HABE and HIBE.

4 Our Generic Transformation

In this section we show how to transform any general-purpose public-key functional encryption scheme into a hierarchical one. Our construction relies on the following building blocks:

1. A general-purpose public-key functional encryption scheme $\mathcal{FE} = (\mathsf{FE.Setup}, \mathsf{FE.KG}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$.
2. A private-key encryption scheme $\mathcal{SKE} = (\mathsf{SKE.KG}, \mathsf{SKE.Enc}, \mathsf{SKE.Dec})$.
3. A puncturable pseudorandom function family $\mathcal{PRF} = (\mathsf{PRF.Gen}, \mathsf{PRF.Eval}, \mathsf{PRF.Punc})$.

Our hierarchical scheme $\mathcal{HFE} = (\mathsf{Setup}, \mathsf{KG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Delegate})$ is defined as follows.

- **The setup algorithm.** On input the security parameter 1^λ the setup algorithm samples and outputs $(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{FE.Setup}(1^\lambda)$.
- **The encryption algorithm.** On input the public key mpk and a message x, the encryption algorithm first samples a PRF key $K \leftarrow \mathsf{PRF.Gen}(1^\lambda)$. Then, it computes and outputs $\mathsf{ct} \leftarrow \mathsf{FE.Enc}(\mathsf{mpk}, (x, K, \bot))$. (Note that the message space of the resulting scheme is thus smaller than that of the original scheme.)
- **The key-generation algorithm.** On input the master secret key msk and a function f, the key-generation algorithm first generates a fresh key pair $(\mathsf{msk}', \mathsf{mpk}') \leftarrow \mathsf{FE.Setup}(1^\lambda)$ and uniformly samples a tag $t \leftarrow \{0,1\}^\lambda$. Then, it uniformly samples $c \leftarrow \{0,1\}^\ell$ where $\ell = \ell(\lambda)$ is the length of an $\mathcal{SKE}$ encryption of an $\mathcal{FE}$ ciphertext.[5] Finally, it computes $\mathsf{sk}_f \leftarrow \mathsf{FE.KG}(\mathsf{msk}, \mathsf{ReEnc}_{f,t,\mathsf{mpk}',c})$, where ReEnc is defined in Figure 1, and outputs $\mathsf{hsk}_f = (\mathsf{sk}_f, \mathsf{msk}')$.
- **The delegation algorithm.** On input a (possibly delegated) functional key of the form $\mathsf{hsk}_{f_i \circ \cdots \circ f_1} = (\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_i}, \mathsf{msk}')$ for some integer $i \geq 1$, and a function f_{i+1}, the delegation algorithm uses the key-generation algorithm described above to compute $(\mathsf{sk}_{f_{i+1}}, \mathsf{msk}'') \leftarrow \mathcal{HFE}.\mathsf{KG}(\mathsf{msk}', f_{i+1})$, and outputs $\mathsf{hsk}_{f_{i+1} \circ \cdots \circ f_1} = (\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_i}, \mathsf{sk}_{f_{i+1}}, \mathsf{msk}'')$.
- **The decryption algorithm.** On input a functional key $\mathsf{hsk}_{f_i \circ \cdots \circ f_1} = (\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_i}, \mathsf{msk}')$ for some integer $i \geq 1$, and a ciphertext ct, the decryption algorithm first sets $\mathsf{ct}_0 = \mathsf{ct}$ and computes $\mathsf{ct}_j \leftarrow \mathsf{FE.Dec}(\mathsf{sk}_{f_j}, \mathsf{ct}_{j-1})$ for $j = 1, \ldots, i$. Then, ct_i is decrypted by using msk' for generating a functional key for the identity function $\mathsf{ID} \in \mathcal{F}$:

  $$w \leftarrow \mathsf{FE.Dec}(\mathsf{FE.KG}(\mathsf{msk}', \mathsf{ID}), \mathsf{ct}_i).$$

  Finally, w is parsed as a triplet $w = (y, \cdot, \cdot)$, of which the first element y is returned as output.

In what follows we first discuss the correctness of our resulting scheme, then discuss its parameters and overhead, and then state and prove its security based on that of its underlying building blocks.

[5] To be accurate, ℓ is also a function of the message space of the scheme and of the specific properties of the master secret key. We refrain from mentioning these implicit parameters to avoid cluttering of notation. We note however that this imposes an a-priori bound on the length of the ciphertext and thus also on the message space of our resulting scheme. Lifting this restriction is an interesting research direction.

$\mathsf{ReEnc}_{f,t,\mathsf{mpk}',c}(x, K, \mathsf{k})$

1. Compute $\mathsf{ct} \leftarrow \mathsf{SKE.Dec}(\mathsf{k}, c)$, $(s, r) = \mathsf{PRF.Eval}(K, t)$, and $K' = \mathsf{PRF.Gen}\left(1^\lambda; s\right)$.
2. If $\mathsf{ct} \neq \bot$ then output ct, and otherwise output $\mathsf{FE.Enc}\left(\mathsf{mpk}', (f(x), K', \bot); r\right)$.

1 The function $\mathsf{ReEnc}_{f,t,\mathsf{mpk}',c}$.

Correctness. The correctness of our scheme follows easily by induction on the delegation depth i. Let $(\mathsf{msk}, \mathsf{mpk}) \leftarrow \mathsf{Setup}(1^\lambda)$, and fix a message $x \in \mathcal{X}_\lambda$ and a sequence of functions $f_1, \ldots, f_i \in \mathcal{F}_\lambda$.

For $i = 1$ the correctness of decrypting a ciphertext $\mathsf{ct}_0 \leftarrow \mathsf{FE.Enc}(\mathsf{mpk}, (x, K_0, \bot))$ using a key $\mathsf{hsk}_{f_1} = (\mathsf{sk}_{f_1}, \mathsf{msk}_1) \leftarrow \mathsf{KG}(\mathsf{msk}, f_1)$ follows from that of the underlying scheme $\mathcal{FE}$. Specifically, the decryption algorithm first computes $\mathsf{ct}_1 \leftarrow \mathsf{FE.Dec}(\mathsf{sk}_{f_1}, \mathsf{ct}_0)$, and by the correctness of $\mathcal{FE}$ with an overwhelming probability it holds that $\mathsf{ct}_1 = \mathsf{FE.Enc}(\mathsf{mpk}_1, (f_1(x), K_1, \bot); r_1)$, where $(s_1, r_1) = \mathsf{PRF.Eval}(K_0, t)$ for some t chosen during the key generation, $K_1 = \mathsf{PRF.Gen}\left(1^\lambda; s_1\right)$, and mpk_1 is a master public key that is sampled together with msk_1. Next, the decryption algorithm decrypts ct_1 with msk_1, and noting that r_1 is pseudorandom given the triplet $(\mathsf{mpk}_1, f_1(x), K_1)$ we can once again rely on the correctness of the underlying scheme $\mathcal{FE}$ and argue that the decryption algorithm outputs $f_1(x)$ with an overwhelming probability.[6]

Assume that the scheme is correct for up to $i-1$ levels of delegation, and consider decrypting a ciphertext $\mathsf{ct}_0 \leftarrow \mathsf{FE.Enc}(\mathsf{mpk}, (x, K_0, \bot))$ using a key $\mathsf{hsk}_{f_i \circ \cdots \circ f_1} = (\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_i}, \mathsf{msk}_i) \leftarrow \mathsf{Delegate}(\mathsf{hsk}_{f_{i-1}}, f_i)$ that is generated using i levels of delegation. Then, the correctness for up to $i-1$ levels guarantees that by repeatedly applying the keys $\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_{i-1}}$ starting with the initial ciphertext ct_0 as prescribed by the decryption algorithm, we obtain with an overwhelming probability a ciphertext $\mathsf{ct}_{i-1} = \mathsf{FE.Enc}(\mathsf{mpk}_{i-1}, ((f_{i-1} \circ \cdots \circ f_1)(x), K_{i-1}, \bot); r_{i-1})$ for some mpk_{i-1}, K_{i-1} and r_{i-1}, where r_{i-1} is pseudorandom given the triplet $(\mathsf{mpk}_{i-1}, (f_{i-1} \circ \cdots \circ f_1)(x), K_{i-1})$. Next, the decryption algorithm computes $\mathsf{ct}_i \leftarrow \mathsf{FE.Dec}(\mathsf{sk}_{f_i}, \mathsf{ct}_{i-1})$, and by the correctness of $\mathcal{FE}$ with an overwhelming probability it holds that $\mathsf{ct}_i = \mathsf{FE.Enc}(\mathsf{mpk}_i, ((f_i \circ \cdots \circ f_1)(x), K_i, \bot); r')$, where $(s_i, r_i) = \mathsf{PRF.Eval}(K_{i-1}, t)$ for some t chosen during the key generation, $K_i = \mathsf{PRF.Gen}\left(1^\lambda; s_{i-1}\right)$, and mpk_i is a master public key that is sampled together with msk_i. Note that again r_i is pseudorandom given the triplet $(\mathsf{mpk}_i, (f_i \circ \cdots \circ f_1)(x), K_i)$. Therefore, when the decryption algorithm decrypts ct_i with msk_i, it outputs $(f_i \circ \cdots \circ f_1)(x)$ with an overwhelming probability.

Parameters and overhead. We now discuss the parameters that govern the properties that are required of the underlying scheme and thus the overhead of our construction. We address two parameters of the hierarchy: The *width* which is the maximal number of delegated keys that are derived from each key at the previous level, and the *depth* which is the maximal number of successive derivations.[7] The functionality and security of our scheme hold for arbitrary and a-priori unbounded width and depth. However, if the underlying scheme is

[6] If we further assume that either the underlying functional encryption scheme is perfectly correct, or that the underlying pseudorandom function produces outputs whose marginal distribution is uniform, the argument significantly simplifies and there is no need to argue that r_1 is pseudorandom given the triplet $(\mathsf{mpk}_1, f_1(x), K_1)$.

[7] One could consider a more fine-grained view of the parameters, e.g. that the maximal width itself depends on the depth of the key. Such analyses follow the same principles presented here.

restricted in some way, then this restriction could propagate through our reduction. For example, if the underlying scheme only supports bounded collusion, then the maximal width will be restricted. Furthermore, since the ReEnc function produces a functional ciphertext with respect to the next level of the hierarchy, certain instantiations could produce a cascading effect that will increase the overhead. We analyze these restrictions below and show that in some cases they can be overcome completely and in others they can be managed.

Define the *compactness parameter* of a (standard) FE scheme, denoted $C(\lambda, S)$, as the computational complexity of encrypting a message of length λ (or some other fixed length which does not depend on S), while allowing to produce functional keys for size S functions. Note that C is also a bound on the length of the ciphertext, and in the currently-known schemes it also governs the complexity of key generation (see Section 2.3 for the currently-known schemes). Then in our construction, the ciphertext encryption complexity at depth i, which we denote by C_i is at most $C_i \leq C(\lambda, C_{i+1} \cdot \mathrm{poly}(\lambda))$. This relation follows immediately from the description of the scheme.

For a scheme which only allows bounded collusion, the compactness is $C(\lambda, S, B)$, where B is the bound on the number of collusions. In this case, the width factors in as well such that for a scheme with width w it holds that $C_i \leq C(\lambda, C_{i+1} \cdot \mathrm{poly}(\lambda), w)$.

In particular, in the known schemes with unbounded collusion [30, 53, 32], the encryption complexity is independent of S and therefore instantiating our construction with such a scheme will support arbitrary polynomial depth and width while keeping the encryption complexity polynomial. In fact, one can show, via a little modification of [5], that any scheme that supports unbounded collusions can be modified using randomized encodings to one where the compactness is independent of S.

For known schemes with bounded collusion, such as those based on public-key encryption [38] or on LWE [37], the compactness is $C(\lambda, S, B) \leq \mathrm{poly}(\lambda) \cdot S \cdot B$, which implies that C_i is bounded by $C_{i+1} \cdot \mathrm{poly}(\lambda) \cdot w$. If we intend to support a total depth d, then unfolding the reduction, the bound we have is $C_0 \leq w^d \cdot \lambda^{O(d)}$. This means that if we wish to keep the encryption complexity polynomial in λ, we can only allow $d = O(1)$ and $w = \mathrm{poly}(\lambda)$. Furthermore, we must know w ahead of time in order to instantiate the parameters of the scheme.

Security. The following theorem captures the security of our resulting scheme. We note that the assumptions stated in the theorem are all known to be implied by the existence of any (selectively-secure) general-purpose public-key functional encryption scheme (see Section 2 for formal descriptions of our building blocks and their known instantiations).

▶ **Theorem 4.1.** *Assuming that (1) $\mathcal{FE}$ is semi-adaptively (resp., selectively) secure (2) $\mathcal{SKE}$ has pseudorandom ciphertexts, and (3) $\mathcal{PRF}$ is a puncturable pseudorandom function family, then $\mathcal{HFE}$ is a semi-adaptively-secure (resp., selectively-secure) hierarchical functional encryption scheme.*

Proof. For ease of exposition we focus here on the case where the underlying scheme $\mathcal{FE}$ is semi-adaptively secure. The proof for the case where $\mathcal{FE}$ is only selectively secure is identical, except for requiring the adversary to provide the challenge messages prior to receiving the public parameters. Let $\mathcal{A}$ be a valid probabilistic polynomial-time adversary (as defined in Section 3). We present a sequence of experiments and upper bound $\mathcal{A}$'s advantage in distinguishing each two consecutive experiments. The first experiment is the experiment $\mathsf{Exp}^{\mathsf{semiHFE}}_{\mathcal{HFE},\mathcal{A}}(\lambda)$ and the last experiment is completely independent of the bit b. This enables

us to prove that there exists a negligible function $\mathsf{neg}(\cdot)$ such that

$$\mathsf{Adv}^{\mathsf{semiHFE}}_{\mathcal{HFE},\mathcal{A}}(\lambda) \stackrel{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Exp}^{\mathsf{semiHFE}}_{\mathcal{HFE},\mathcal{A}}(\lambda)=1\right]-\frac{1}{2}\right| \leq \mathsf{neg}(\lambda)$$

for all sufficiently large $\lambda \in \mathbb{N}$.

How to read this proof. To read our proof, one starts from the first hybrid and proceeds in order to the next, each adjacent hybrid is shown to be computationally indistinguishable from its predecessor. When a loop is encountered, this means that a sequence of hybrids is now being defined, one hybrid for each "iteration" of the loop. The hybrid defined in the first iteration needs to be indistinguishable from the last hybrid before the loop, and all hybrids except the first need to be indistinguishable from the hybrid of the previous iteration. In a *nested loop*, each iteration of the external loop represents a generation of many hybrids, as many as the internal loop generates. In such case, in the first iteration of the external loop, and the first iteration of the internal loop, the hybrid being defined needs to be indistinguishable from the one preceding the loop. However, in the next execution of the external loop, the first iteration of the internal needs to be indistinguishable with the *last* iteration of the internal loop that have been carried out in the previous iteration of the external loop. For example, say that the external loop iterates for $i = 1, \ldots, S$ and the internal loop iterates for $j = 1, \ldots, T$. Then what we prove for $\mathcal{H}^{(i,j)}$ is that: $\mathcal{H}^{(1,1)}$ is indistinguishable from the last hybrid before the loop, $\mathcal{H}^{(i,1)}$ for $i > 1$ is indistinguishable from $\mathcal{H}^{(i-1,T)}$, and for $i, j > 1$ that $\mathcal{H}^{(i,j)}$ is indistinguishable from $\mathcal{H}^{(i,j-1)}$.

In order to explain the purpose of the different steps in the proof, we also include *invariants* which are properties of the distribution of the current experiment. The invariant holds *only* at that point in the proof where it appears and does not necessarily hold in following hybrids. An invariant inside a loop holds whenever the flow of the proof reaches that point in the loop. Namely, going back to our nested loop example from above, an invariant that appears after the "for $i = 1, \ldots, S$" statement, holds for the experiment immediately preceding the loop, and for all hybrids $\mathcal{H}^{(i,T)}$, except $\mathcal{H}^{(S,T)}$. An invariant that appears after the "for $j = 1, \ldots, T$" statement, should hold for the hybrid immediately preceding the loop, as well as for all $\mathcal{H}^{(i,T)}$, except $\mathcal{H}^{(S,T)}$.

We advise the reader to read our proof as if it was an execution of a computer program. We believe that while this proof writing method is still not very widely used, it is quite beneficial in writing complicated proofs, and will find additional uses. In what follows we first describe the notation used throughout the proof, and then describe the experiments.

Notation. Let $Q = Q(\lambda)$ denote a polynomial upper bound on the number of queries that are made by $\mathcal{A}$ in the experiment $\mathsf{Exp}^{\mathsf{semiHFE}}_{\mathcal{HFE},\mathcal{A}}(\lambda)$. We denote these queries by $\{(f_i, \mathsf{parent}_i, \mathsf{mode}_i)\}_{i\in[Q]}$ and we also consider an implicit "zeroth" query which generates the master key pair $(\mathsf{msk}, \mathsf{mpk})$ of the scheme. This allows us to define the *depth* of the ith query, denoted $\mathsf{d}(i)$, s.t. $\mathsf{d}(0) = 0$ and $\mathsf{d}(i) = \mathsf{d}(\mathsf{p}(i)) + 1$ for $i > 0$, where we use $\mathsf{p}(i)$ as shorthand for parent_i. Thus we can view $\mathcal{A}$'s queries as a tree rooted by the zeroth query, where each query $(f_i, \mathsf{parent}_i, \mathsf{mode}_i)$ is the child of the query $\mathsf{p}(i)$ and has depth $\mathsf{d}(i)$ in the tree.

For any query $i \in \{0, \ldots, Q\}$, we define a function $\tilde{f}_i$ as follows: $\tilde{f}_0$ is the identity function, and for all $i > 0$ we define $\tilde{f}_i = f_i \circ \tilde{f}_{\mathsf{p}(i)}$. In other words, the ith query $(f_i, \mathsf{parent}_i, \mathsf{mode}_i)$ generates a delegated key that allows to compute the function $\tilde{f}_i(x)$ given an encryption of x. We say that the ith query is *observable* if $\tilde{f}_i(x^*_0) = \tilde{f}_i(x^*_1)$, and *unobservable* otherwise. We note that if the ith query is unobservable then necessarily $\mathsf{mode}_i = \mathsf{StoreKey}$.

We let $(\mathsf{msk}_i, \mathsf{mpk}_i)$ denote the key pair generated by the challenger while answering the ith query, and let $(\mathsf{msk}_0, \mathsf{mpk}_0)$ be the master key pair $(\mathsf{msk}, \mathsf{mpk})$ that is generated by the setup algorithm. Similarly, we let t_i denote the tag that is sampled while answering the ith query.

We denote by x_0^* and x_1^* the challenge messages that are chosen by $\mathcal{A}$, and by K^* the PRF key that is used for computing the challenge ciphertext. We further define $K_0^* = K^*$, and for all $i > 0$ we define K_i^*, s_i^*, and r_i^* as follows: $(s_i^*, r_i^*) = \mathsf{PRF.Eval}(K^*_{\mathsf{p}(i)}, t_i)$, and $K_i^* = \mathsf{PRF.Gen}(1^\lambda; s_i^*)$. Note that these are exactly the values that are computed by the ReEnc function produced in the ith query, when evaluated on the challenge ciphertext.

Finally, throughout the proof we find it convenient to denote by $\$$ a fresh value that is sampled uniformly and independently of all other existing values.

Experiment $\mathcal{H}_0$. This is the experiment $\mathsf{Exp}^{\mathsf{semiHFE}}_{\mathcal{HFE},\mathcal{A}}(\lambda)$ (see Section 3).

Experiment $\mathcal{H}_1$. This experiment is obtained from the experiment $\mathcal{H}_0$ by having the challenger sample in advance the tags and the key pairs that are used for replying to $\mathcal{A}$'s queries. In fact, we will sample these values in a redundant manner so that we prepare several such triplets for each query, and the choice of which triplet to use is determined by the depth of the query. We thus have the following claim:

Specifically, at the beginning of the experiment, for all $i, d \in [Q]$ the challenger samples $t^{(\mathrm{o})}_{i,d}, t^{(\mathrm{u})}_{i,d} \leftarrow \{0,1\}^\lambda$ and $(\mathsf{msk}_{i,d}, \mathsf{mpk}_{i,d}) \leftarrow \mathsf{FE.Setup}(1^\lambda)$. Then, the experiment proceeds exactly as in $\mathcal{H}_1$, and whenever the challenger needs to sample t_i and $(\mathsf{msk}_i, \mathsf{mpk}_i)$ for replying to the ith query, it will use $t_i = t^{(\mathrm{o})}_{i,\mathsf{d}(i)}$ if i is an observable query, and $t_i = t^{(\mathrm{u})}_{i,\mathsf{d}(i)}$ otherwise. It will further use $(\mathsf{msk}_i, \mathsf{mpk}_i) = (\mathsf{msk}_{i,\mathsf{d}(i)}, \mathsf{mpk}_{i,\mathsf{d}(i)})$.

looking ahead, this experiment allows the challenger to know in advance, for every possible depth, a polynomial superset of the tags and key pairs that will be produced for replying to queries of this depth. The view of the adversary in this experiment is distributed identically to its view in the experiment $\mathcal{H}_0$, yielding the following observation:

▶ **Observation 4.2.** *For all $\lambda \in \mathbb{N}$ it holds that*

$$\Pr[\mathcal{H}_0(\lambda) = 1] = \Pr[\mathcal{H}_1(\lambda) = 1]\,.$$

Experiment $\mathcal{H}_2$. This experiment is obtained from the experiment $\mathcal{H}_1$ as follows. After the generation of the tags $t^{(\mathrm{o})}_{i,d}$ and $t^{(\mathrm{u})}_{i,d}$, and before interacting with the adversary, the challenger checks if any of the values $t^{(\mathrm{o})}_{i,d}$ or $t^{(\mathrm{u})}_{i,d}$ for some $(i,d) \in [Q]^2$ appears more than once. In such case the output of the experiment is defined as $\perp$, and otherwise the experiment is identical to the experiment $\mathcal{H}_1$. A standard union bound implies that the experiments $\mathcal{H}_1$ and $\mathcal{H}_2$ differ with probability at most $2(Q+1)^4 \cdot 2^{-\lambda} = \mathsf{neg}(\lambda)$, yielding the following observation:

▶ **Observation 4.3.** *For all $\lambda \in \mathbb{N}$ it holds that*

$$|\Pr[\mathcal{H}_1(\lambda) = 1] - \Pr[\mathcal{H}_2(\lambda) = 1]| \leq \frac{2(Q+1)^4}{2^\lambda}.$$

Experiment $\mathcal{H}_3$. This experiment is obtained from the experiment $\mathcal{H}_2$ by sampling a sequence $\mathsf{k}_0, \ldots, \mathsf{k}_{Q-1} \leftarrow \mathsf{SKE.KG}(1^\lambda)$ of symmetric keys (one for each possible depth – recall that Q is always an upper bound on the maximal depth), and modifying the symmetric ciphertext c that is generated by the key-generation algorithm when replying to each query as

follows: When replying to the ith query $(f_i, \mathsf{parent}_i, \mathsf{mode}_i)$, instead of sampling c uniformly, the key-generation algorithm computes

$$\boxed{c_i = \mathsf{SKE.Enc}\left(\mathsf{k}_{\mathsf{d}(i)-1}, \mathsf{ct}_i; \$\right)}$$

where $\mathsf{ct}_i = \mathsf{FE.Enc}(\mathsf{mpk}_i, (\tilde{f}_i(x_b^*), K_i^*, \bot); r_i^*)$ (recall that throughout the proof we find it convenient to denote by $\$$ a fresh value that is sampled uniformly and independently of all other existing values).

Note that ct_i is exactly the same as the "ct_i" value that is computed in the process of decrypting the challenge ciphertext using the ith functional key (and is also computed as an intermediate value when decrypting the challenge ciphertext with any descendant of the ith key). See the decryption algorithm above.

It thus makes sense to extend our notation and denote the challenge ciphertext by ct_0 (as in the decryption algorithm). Note that while ct_0 is encrypted with true randomness and includes a properly generated PRF key, all other ct_i's are encrypted using pseudorandomness and contain PRF keys that were generated pseudorandomly. We further say that ct_i is observable if the ith query is an observable query and unobservable otherwise.

To see why the adversary's view in $\mathcal{H}_3$ is indistinguishable from $\mathcal{H}_2$, we note that in $\mathcal{H}_3$, the symmetric keys $\mathsf{k}_0, \ldots, \mathsf{k}_{Q-1}$ are used only for generating the c_i's. In other words, this experiment can be carried out given only access to an encryption oracle $\mathsf{SKE.Enc}(\mathsf{k}_d, \cdot)$ for each $d \in \{0, \ldots, Q-1\}$ (instead of explicit access to the actual keys $\mathsf{k}_0, \ldots, \mathsf{k}_{Q-1}$). This enables us to use the ciphertext pseudorandomness of $\mathcal{SKE}$ to prove computational indistinguishability from $\mathcal{H}_2$, yielding the following claim in a rather straightforward manner:

▶ **Claim 4.4.** *Assuming that $\mathcal{SKE}$ has pseudorandom ciphertexts, there exists a negligible function $\mathsf{neg}(\cdot)$ such that*

$$|\Pr[\mathcal{H}_2(\lambda) = 1] - \Pr[\mathcal{H}_3(\lambda) = 1]| \leq \mathsf{neg}(\lambda)$$

for all sufficiently large $\lambda \in \mathbb{N}$.

For $d = 0, \ldots, Q$:

▶ **Invariant 4.5.** *In the previous experiment, it should hold that all ciphertexts ct_i that correspond to* unobservable *queries (i.e., queries for which $\tilde{f}_i(x_0^*) \neq \tilde{f}_i(x_1^*)$) such that $\mathsf{d}(i) < d$ are of the form $\mathsf{FE.Enc}(\mathsf{mpk}_i, (\bot, K_i^*, \mathsf{k}_{\mathsf{d}(i)}); \$)$, and all such ciphertext such that $\mathsf{d}(i) = d$ are of the form $\mathsf{ct}_i = \mathsf{FE.Enc}(\mathsf{mpk}_i, (\tilde{f}_i(x_b^*), K_i^*, \bot); \$)$. Further, if $\mathsf{d}(i) \leq d$ then $K_i^* = \mathsf{PRF.Gen}(1^\lambda; \$)$. More specifically:*

- *If i is such that $\mathsf{d}(i) < d$ and $\tilde{f}_i(x_0^*) \neq \tilde{f}_i(x_1^*)$, then it holds that*

  $$\mathsf{ct}_i = \mathsf{FE.Enc}\left(\mathsf{mpk}_i, \left(\bot, K_i^*, \mathsf{k}_{\mathsf{d}(i)}\right); \$\right)$$

  and $K_i^* = \mathsf{PRF.Gen}(1^\lambda; \$)$.
- *If i is such that $\mathsf{d}(i) = d$ and $\tilde{f}_i(x_0^*) \neq \tilde{f}_i(x_1^*)$, then it holds that*

  $$\mathsf{ct}_i = \mathsf{FE.Enc}\left(\mathsf{mpk}_i, \left(\tilde{f}_i(x_b^*), K_i^*, \bot\right); \$\right)$$

  and $K_i^* = \mathsf{PRF.Gen}(1^\lambda; \$)$.
- *If i is such that $\mathsf{d}(i) > d$ or $\tilde{f}_i(x_0^*) = \tilde{f}_i(x_1^*)$, then it holds that*

  $$\mathsf{ct}_i = \mathsf{FE.Enc}\left(\mathsf{mpk}_i, \left(\tilde{f}_i(x_b^*), K_i^*, \bot\right); r_i^*\right)$$

  and $K_i^* = \mathsf{PRF.Gen}(1^\lambda; s_i^*)$.

We note that this indeed holds for $d = 0$ in experiment $\mathcal{H}_3$.

For $i = 0, \ldots, Q$:

Experiment $\mathcal{H}_4^{(i,d)}$. In this experiment, the challenger changes the way ct_i is computed as follows. Before generating ct_i, the challenger checks if both $\mathsf{d}(i) = d$ and ct_i is unobservable ($\tilde{f}_i(x_0^*) \neq \tilde{f}_i(x_1^*)$). If both conditions hold then it sets

$$\mathsf{ct}_i = \mathsf{FE.Enc}(\mathsf{mpk}_i, (\bot, K_i^*, \mathsf{k}_d); \$) \ .$$

Otherwise ct_i is computed as in the previous experiment.

To see why the adversary's view in this experiment is indistinguishable from the previous hybrid, we note that for any child of i, i.e., j such that $\mathsf{p}(j) = i$,

$$\mathsf{ReEnc}_{f_j,t_j,\mathsf{mpk}_j,c_j}(\tilde{f}_i(x_b^*), K_i^*, \bot) = \mathsf{ReEnc}_{f_j,t_j,\mathsf{mpk}_j,c_j}(\bot, K_i^*, \mathsf{k}_d) = \mathsf{ct}_j \ .$$

This is because necessarily $\mathsf{d}(j) = d + 1 > d$ and due to Invariant 4.5. Thus, the security of the $(\mathsf{msk}_{i,d}, \mathsf{mpk}_{i,d})$ key pair guarantees that this hybrid is indistinguishable from the previous one: Since ct_i is unobservable, then necessarily the adversary cannot access $\mathsf{msk}_{i,d}$ directly, but rather only via further delegation (i.e., via functional keys to $\mathsf{ReEnc}_{f_j,t_j,\mathsf{mpk}_j,c_j}$). This yields the following claim in rather straightforward manner:

▶ **Claim 4.6.** *Assuming that $\mathcal{FE}$ is semi-adaptively secure, there exists a negligible function* $\mathsf{neg}(\cdot)$ *such that*

$$\left|\Pr\left[\mathcal{H}_4^{(0,0)}(\lambda) = 1\right] - \Pr[\mathcal{H}_3(\lambda) = 1]\right| \leq \mathsf{neg}(\lambda)$$

and

$$\left|\Pr\left[\mathcal{H}_4^{(i,d)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}_4^{(i-1,d)}(\lambda) = 1\right]\right| \leq \mathsf{neg}(\lambda)$$

for all $d \in \{0, \ldots, Q\}$ and $i \in \{1, \ldots, Q\}$, and for all sufficiently large $\lambda \in \mathbb{N}$.

End For i.

▶ **Invariant 4.7.** *In the previous experiment, it should hold that all ciphertexts ct_i corresponding to unobservable queries such that $\mathsf{d}(i) \leq d$ are of the form $\mathsf{FE.Enc}(\mathsf{mpk}_i, (\bot, K_i^*, \mathsf{k}_{\mathsf{d}(i)}); \$)$ and further $K_i^* = \mathsf{PRF.Gen}(1^\lambda, \$)$.*

Recall that our goal is to restore Invariant 4.5 for value $(d + 1)$. To this end, we next need to replace r_j^* and s_j^* for all j such that $\mathsf{d}(j) = d + 1$, with random values (rather than values that are generated from $K_{\mathsf{p}(j)}^*$).

For $j = 0, \ldots, Q$:

▶ **Invariant 4.8.** *This is similar to Invariant 4.7, but for all ciphertexts $\mathsf{ct}_{j'}$ corresponding to unobservable queries such that $j' < j$ and $\mathsf{d}(j') = d + 1$ it holds that $r_{j'}^*$ and $s_{j'}^*$ had already been replaced with random.*

For $i = 0, \ldots, Q$:

Experiment $\mathcal{H}_5^{(i,j,d)}$. In this hybrid, we again change ct_i as follows. If ct_i is unobservable and $\mathsf{d}(i) = d$, then define $K_i^* = \mathsf{PRF.Gen}(1^\lambda, \$)$ (as before), $K_i^{\circledast} = \mathsf{PRF.Punc}(K_i^*, t_{j,d+1}^{(\mathrm{u})})$, $y_{i,j,d+1} = \mathsf{PRF.Eval}(K_i^*, t_{j,d+1}^{(\mathrm{u})})$. We now set:

$$\mathsf{ct}_i = \mathsf{FE.Enc}\left(\mathsf{mpk}_i, \left(\bot, \left(K_i^{\circledast}, (t_{j,d+1}^{(\mathrm{u})}, y_{i,j,d+1})\right), \mathsf{k}_{\mathsf{d}(i)}\right); \$\right) .$$

Namely, we replace the PRF key with a punctured key at the point $t_{j,d+1}^{(\mathrm{u})}$, and supply the value at that point[8]. We note that the functionality of $\mathsf{PRF.Eval}((K_i^{\circledast}, (t_{j,d+1}^{(\mathrm{u})}, y_{i,j,d+1})), \cdot)$ is identical to $\mathsf{PRF.Eval}(K_i^*, \cdot)$. The security of the key pair $(\mathsf{msk}_{i,d}, \mathsf{mpk}_{i,d})$ guarantees the indistinguishability of this hybrid (again relying on ct_i being unobservable and thus $\mathsf{msk}_{i,d}$ is not given to the adversary). This yields the following claim in rather straightforward manner:

▶ **Claim 4.9.** *Assuming that $\mathcal{FE}$ is semi-adaptively secure, there exists a negligible function $\mathsf{neg}(\cdot)$ such that*

$$\left|\Pr\left[\mathcal{H}_5^{(0,0,d)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}_4^{Q,d}(\lambda) = 1\right]\right| \leq \mathsf{neg}(\lambda)$$

for all $d \in \{0, \ldots, Q\}$, and

$$\left|\Pr\left[\mathcal{H}_5^{(i,j,d)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}_5^{(i-1,j,d)}(\lambda) = 1\right]\right| \leq \mathsf{neg}(\lambda)$$

for all $d, j \in \{0, \ldots, Q\}$ and $i \in \{1, \ldots, Q\}$, and for all sufficiently large $\lambda \in \mathbb{N}$.

End For i.

▶ **Invariant 4.10.** *In the current experiment, it holds that the PRF key for all depth-d ciphertexts which are unobservable had been punctured at point $t_{j,d+1}^{(u)}$, namely at the point on which it will be evaluated if indeed ct_j is of level $d+1$.*

For $i = 0, \ldots, Q$:

Experiment $\mathcal{H}_6^{(i,j,d)}$. In this hybrid, we again change ct_i in the case where ct_i is unobservable and $\mathsf{d}(i) = d$. The change from the previous experiment is only that now $y_{i,j,d+1} \leftarrow \$$, namely sampled randomly. We notice that already in the previous hybrid we never use K^* for unobservable queries, only the respective $K^{\circledast}$ and y values. Therefore swapping the y value to a completely random will be indistinguishable to the adversary by the punctured PRF property. This yields the following claim in rather straightforward manner:

▶ **Claim 4.11.** *Assuming that $\mathcal{PRF}$ is a puncturable pseudorandom function, there exists a negligible function $\mathsf{neg}(\cdot)$ such that*

$$\left|\Pr\left[\mathcal{H}_6^{(0,j,d)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}_5^{Q,j,d}(\lambda) = 1\right]\right| \leq \mathsf{neg}(\lambda)$$

[8] As discussed in Section 2.1, we find it natural to consider an "augmented" evaluation algorithm that outputs a pre-determined value at the punctured point. That is, the augmented evaluation algorithm is given an augmented key $(K_i^{\circledast}, (t_{j,d+1}^{(\mathrm{u})}, y_{i,j,d+1}))$, where $K_i^{\circledast}$ is punctured at $t_{j,d+1}^{(\mathrm{u})}$, and given an input t it outputs $\mathsf{PRF.Eval}_{K_i^{\circledast}}(t)$ if $t \neq t_{j,d+1}^{(\mathrm{u})}$, and it outputs $y_{i,j,d+1}$ if $t = t_{j,d+1}^{(\mathrm{u})}$.

and

$$\left|\Pr\left[\mathcal{H}_6^{(i,j,d)}(\lambda)=1\right]-\Pr\left[\mathcal{H}_6^{(i-1,j,d)}(\lambda)=1\right]\right|\leq \mathsf{neg}(\lambda)$$

for all $d,j\in\{0,\ldots,Q\}$ *and* $i\in\{1,\ldots,Q\}$*, and for all sufficiently large* $\lambda\in\mathbb{N}$*.*

End For i.

▶ **Invariant 4.12.** *In the current experiment, it holds that the PRF key for all depth-d ciphertexts which are unobservable had been punctured at point* $t_{j,d+1}$*, and further that the punctured value had been substituted with random.*

Experiment $\mathcal{H}_7^{(j,d)}$. In this hybrid, we (finally) change the way ct_j is generated in the case where ct_j is unobservable and $\mathsf{d}(j)=d+1$ (if these conditions don't hold then we proceed as in the previous experiment). In particular, we change the way the randomness for ct_j and K_j^* is generated. Note that if ct_j is unobservable then it must be the case that $\mathsf{ct}_{\mathsf{p}(j)}$ is also unobservable (since $\tilde{f}_j=f_j\circ\tilde{f}_{\mathsf{p}(j)}$). In the previous experiment, we had

$$(s_j^*,r_j^*)=\mathsf{PRF.Eval}\left(\left(K^{\circledast}_{\mathsf{p}(j)},(t^{(\mathrm{u})}_{j,d+1},y_{\mathsf{p}(j),j,d+1})\right),t^{(\mathrm{u})}_{j,d+1}\right)\ .$$

We now define instead $(s'_j,r'_j)=y_{\mathsf{p}(j),j,d+1}$. We set $K_j^*=\mathsf{PRF.Gen}(1^\lambda,s'_j)$ and

$$\mathsf{ct}_j=\mathsf{FE.Enc}\left(\mathsf{mpk}_j,\left(\tilde{f}_j(x_b^*),K_j^*,\bot\right);r'_j\right)\ .$$

The view of the adversary here remains exactly the same, since $(s'_j,r'_j)=(s_j^*,r_j^*)$. However, conceptually this means that (s_j^*,r_j^*) are detached from the value that is embedded in $\mathsf{ct}_{\mathsf{p}(i)}$. As we will see in the next experiment, we will remove $y_{i,j,d+1}$ from ct_i, but (s'_j,r'_j) will still be well defined. This yields the following observation:

▶ **Observation 4.13.** *For all* $\lambda\in\mathbb{N}$ *it holds that*

$$\Pr\left[\mathcal{H}_7^{(j,d)}(\lambda)=1\right]=\Pr\left[\mathcal{H}_6^{Q,j,d}(\lambda)=1\right]$$

for all $d,j\in\{0,\ldots,Q\}$*.*

For $i=0,\ldots,Q$:

Experiment $\mathcal{H}_8^{(i,j,d)}$. In this hybrid, we again change ct_i in the case where ct_i is unobservable and $\mathsf{d}(i)=d$. We will now undo the puncturing of the PRF keys.

$$\mathsf{ct}_i=\mathsf{FE.Enc}\left(\mathsf{mpk}_i,\left(\bot,K_i^*,\mathsf{k}_{\mathsf{d}(i)}\right);\$\right)\ .$$

Indistinguishability holds since in all positions except $t^{(\mathrm{u})}_{j,d+1}$ the new and old keys, K_i^* and $\left(K^{\circledast}_i,(t_{j,d+1},y_{i,j,d+1})\right)$ are functionally equivalent. Furthermore, the function $\mathsf{PRF.Eval}$ is never evaluated at $t^{(\mathrm{u})}_{j,d+1}$ (since if ct_j is unobservable then (r'_j,s'_j) are used instead of (r_j^*,s_j^*)). The functional encryption security of $(\mathsf{msk}_{i,d},\mathsf{mpk}_{i,d})$ therefore implies indistinguishability. This yields the following claim in rather straightforward manner:

▶ **Claim 4.14.** *Assuming that* $\mathcal{FE}$ *is semi-adaptively secure, there exists a negligible function* $\mathsf{neg}(\cdot)$ *such that*

$$\left|\Pr\left[\mathcal{H}_8^{(0,j,d)}(\lambda)=1\right]-\Pr\left[\mathcal{H}_7^{j,d}(\lambda)=1\right]\right|\leq \mathsf{neg}(\lambda)$$

for all $d, j \in \{0, \ldots, Q\}$*, and*

$$\left|\Pr\left[\mathcal{H}_8^{(i,j,d)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}_8^{(i-1,j,d)}(\lambda) = 1\right]\right| \leq \mathsf{neg}(\lambda)$$

for all $d, j \in \{0, \ldots, Q\}$ *and* $i \in \{1, \ldots, Q\}$*, and for all sufficiently large* $\lambda \in \mathbb{N}$*.*

End For i.

The proof of the following claim is almost identical to that of Claim 4.9 and is therefore omitted:

▶ **Claim 4.15.** *Assuming that* $\mathcal{FE}$ *is semi-adaptively secure, there exists a negligible function* $\mathsf{neg}(\cdot)$ *such that*

$$\left|\Pr\left[\mathcal{H}_8^{(Q,j,d)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}_5^{(0,j+1,d)}(\lambda) = 1\right]\right| \leq \mathsf{neg}(\lambda)$$

for all $d \in \{0, \ldots, Q\}$ *and* $j \in \{0, \ldots, Q-1\}$*, and for all sufficiently large* $\lambda \in \mathbb{N}$*.*

End For j.

The proof of the following claim is almost identical to that of Claim 4.6 and is therefore omitted:

▶ **Claim 4.16.** *Assuming that* $\mathcal{FE}$ *is semi-adaptively secure, there exists a negligible function* $\mathsf{neg}(\cdot)$ *such that*

$$\left|\Pr\left[\mathcal{H}_8^{(Q,Q,d)}(\lambda) = 1\right] - \Pr\left[\mathcal{H}_4^{(0,d+1)}(\lambda) = 1\right]\right| \leq \mathsf{neg}(\lambda)$$

for all $d \in \{0, \ldots, Q-1\}$*, and for all sufficiently large* $\lambda \in \mathbb{N}$*.*

End For d.

We now notice that the proof is practically finished, since the last hybrid $\mathcal{H}_8^{(Q,Q,Q)}$ is completely independent of the bit b. To see this, note that the only values that depend on b in the experiment are the values $\tilde{f}_i(x_b^*)$ that appear inside the ciphertexts ct_i (in particular inside the challenge ciphertext $\mathsf{ct}^* = \mathsf{ct}_0$). We first point out that the value $\tilde{f}_i(x_b^*)$ is in fact *independent* of b in observable ciphertexts, since by definition $\tilde{f}_i(x_0^*) = \tilde{f}_i(x_1^*)$. As for unobservable ciphertexts, in $\mathcal{H}_8^{(Q,Q,Q)}$ none of them contains $\tilde{f}_i(x_b^*)$ at all, as this value had been replaced by $\perp$. This yields the following observation:

▶ **Observation 4.17.** *For all* $\lambda \in \mathbb{N}$ *it holds that*

$$\Pr\left[\mathcal{H}_8^{(Q,Q,Q)}(\lambda) = 1\right] = \frac{1}{2}\ .$$

We presented a sequence of polynomially-many experiments starting with the experiment $\mathcal{H}_0 = \mathsf{Exp}^{\mathsf{semiHFE}}_{\mathcal{HFE},\mathcal{A}}$ and ending with the experiment $\mathcal{H}_8^{(Q,Q,Q)}$ which is completely independent of the bit b. We showed that the output distributions of each two consecutive experiments are negligibly close, which implies that there exists a negligible function $\mathsf{neg}(\cdot)$ such that

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{semiHFE}}_{\mathcal{HFE},\mathcal{A}}(\lambda) &\stackrel{\mathsf{def}}{=} \left|\Pr\left[\mathsf{Exp}^{\mathsf{semiHFE}}_{\mathcal{HFE},\mathcal{A}}(\lambda) = 1\right] - \frac{1}{2}\right| \\
&= \left|\Pr\left[\mathcal{H}_0(\lambda) = 1\right] - \Pr\left[\mathcal{H}_8^{(Q,Q,Q)}(\lambda) = 1\right]\right| \\
&\leq \mathsf{neg}(\lambda)
\end{aligned}$$

for all sufficiently large $\lambda \in \mathbb{N}$, as required. ◀

References

1. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *Advances in Cryptology – EUROCRYPT'10*, pages 553–572, 2010.
2. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *Advances in Cryptology – CRYPTO'10*, pages 98–115, 2010.
3. Shweta Agrawal, Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption: New perspectives and lower bounds. In *Advances in Cryptology – CRYPTO'13*, pages 500–518, 2013.
4. Prabhanjan Ananth, Dan Boneh, Sanjam Garg, Amit Sahai, and Mark Zhandry. Differing-inputs obfuscation and applications. Cryptology ePrint Archive, Report 2013/689, 2013.
5. Prabhanjan Ananth, Zvika Brakerski, Gil Segev, and Vinod Vaikuntanathan. From selective to adaptive security in functional encryption. In *Advances in Cryptology – CRYPTO'15*, pages 657–677, 2015.
6. Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In *Advances in Cryptology – CRYPTO'15*, pages 308–326, 2015.
7. Gilad Asharov and Gil Segev. Limits on the power of indistinguishability obfuscation and functional encryption. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, pages 191–209, 2015.
8. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. *Journal of the ACM*, 59(2):6, 2012.
9. Mihir Bellare and Adam O'Neill. Semantically-secure functional encryption: Possibility results, impossibility results and the quest for a general definition. In *Proceedings of the 12th International Conference on Cryptology and Network Security*, pages 218–234, 2013.
10. Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, pages 171–190, 2015.
11. Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In *Advances in Cryptology – EUROCRYPT'04*, pages 223–238, 2004.
12. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology – EUROCRYPT'05*, pages 440–456, 2005.
13. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003. Preliminary version in *Advances in Cryptology – CRYPTO'01*, pages 213–229, 2001.
14. Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In *Advances in Cryptology – EUROCRYPT'14*, pages 533–556, 2014.
15. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In *Proceedings of the 8th Theory of Cryptography Conference*, pages 253–273, 2011.
16. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: a new vision for public-key cryptography. *Communications of the ACM*, 55(11):56–64, 2012.
17. Dan Boneh and Brent Waters. Constrained pseudorandom functions and their applications. In *Advances in Cryptology - ASIACRYPT'13*, pages 280–300, 2013.
18. Xavier Boyen and Brent Waters. Anonymous hierarchical identity-based encryption (without random oracles). In *Advances in Cryptology – CRYPTO'06*, pages 290–307, 2006.

19 Elette Boyle, Kai-Min Chung, and Rafael Pass. On extractability obfuscation. In *Proceedings of the 11th Theory of Cryptography Conference*, pages 52–73, 2014.

20 Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In *Proceedings of the 17th International Conference on Practice and Theory in Public-Key Cryptography*, pages 501–519, 2014.

21 Zvika Brakerski, Ilan Komargodski, and Gil Segev. Multi-input functional encryption in the private-key setting: Stronger security from weaker assumptions. In *Advances in Cryptology – EUROCRYPT'16*, pages 852–880, 2016.

22 Zvika Brakerski and Gil Segev. Function-private functional encryption in the private-key setting. In *Proceedings of the 12th Theory of Cryptography Conference*, pages 306–324, 2015.

23 Zvika Brakerski and Gil Segev. Hierarchical functional encryption. Cryptology ePrint Archive, Report 2015/1011, 2015.

24 David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. *Journal of Cryptology*, 25(4):601–639, 2012.

25 Nishanth Chandran, Vipul Goyal, Aayush Jain, and Amit Sahai. Functional encryption: Decentralised and delegatable. Cryptology ePrint Archive, Report 2015/1017, 2015.

26 Jie Chen and Hoeteck Wee. Semi-adaptive attribute-based encryption and improved delegation for Boolean formula. In *Proceedings of the 9th International Conference on Security and Cryptography for Networks*, pages 277–297, 2014.

27 Clifford Cocks. An identity based encryption scheme based on quadratic residues. In *Proceedings of the 8th IMA International Conference on Cryptography and Coding*, pages 360–363, 2001.

28 Ronald Cramer, Ivan Damgård, and Yuval Ishai. Share conversion, pseudorandom secret-sharing and applications to secure computation. In *Proceedings of the 2nd Theory of Cryptography Conference*, pages 342–362, 2005.

29 Angelo De Caro, Vincenzo Iovino, Abhishek Jain, Adam O'Neill, Omer Paneth, and Giuseppe Persiano. On the achievability of simulation-based security for functional encryption. In *Advances in Cryptology – CRYPTO'13*, pages 519–535, 2013.

30 Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science*, pages 40–49, 2013.

31 Sanjam Garg, Craig Gentry, Shai Halevi, and Daniel Wichs. On the implausibility of differing-inputs obfuscation and extractable witness encryption with auxiliary input. In *Advances in Cryptology – CRYPTO'14*, pages 518–535, 2014.

32 Sanjam Garg, Craig Gentry, Shai Halevi, and Mark Zhandry. Functional encryption without obfuscation. In *Proceedings of the 1th Theory of Cryptography Conference*, pages 480–511, 2016.

33 Craig Gentry and Shai Halevi. Hierarchical identity based encryption with polynomially many levels. In *Proceedings of the 6th Theory of Cryptography Conference*, pages 437–456, 2009.

34 Craig Gentry and Alice Silverberg. Hierarchical ID-based cryptography. In *Advances in Cryptology – ASIACRYPT'02*, pages 548–566, 2002.

35 Oded Goldreich. *Foundations of Cryptography – Volume 2: Basic Applications*. Cambridge University Press, 2004.

36 Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.

37 Shafi Goldwasser, Yael Kalai, Raluca Ada Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 555–564, 2013.

38 Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In *Advances in Cryptology – CRYPTO'12*, pages 162–179, 2012.

39 Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 545–554, 2013.

40 Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In *Advances in Cryptology – EUROCRYPT'02*, pages 466–481, 2002.

41 Aggelos Kiayias, Stavros Papadopoulos, Nikos Triandopoulos, and Thomas Zacharias. Delegatable pseudorandom functions and applications. In *Proceedings of the 20th Annual ACM Conference on Computer and Communications Security*, pages 669–684, 2013.

42 Ilan Komargodski, Gil Segev, and Eylon Yogev. Functional encryption for randomized functionalities in the private-key setting from minimal assumptions. In *Proceedings of the 12th Theory of Cryptography Conference*, pages 352–377, 2015.

43 Allison B. Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *Advances in Cryptology – EUROCRYPT'10*, pages 62–91, 2010.

44 Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *Poceedings of the 7th Theory of Cryptography Conference*, pages 455–479, 2010.

45 Allison B. Lewko and Brent Waters. Unbounded HIBE and attribute-based encryption. In *Advances in Cryptology – EUROCRYPT'11*, pages 547–567, 2011.

46 Allison B. Lewko and Brent Waters. Why proving HIBE systems secure is difficult. In *Advances in Cryptology – EUROCRYPT'14*, pages 58–76, 2014.

47 Pratyay Mukherjee and Daniel Wichs. Two round multiparty computation via multi-key FHE. In *Advances in Cryptology – EUROCRYPT'16*, pages 735–763, 2016.

48 Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010.

49 Amit Sahai and Brent Waters. Slides on functional encryption. Available at `http://www.cs.utexas.edu/~bwaters/presentations/files/functional.ppt`, 2008.

50 Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 475–484, 2014.

51 Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology – CRYPTO'84*, pages 47–53, 1984.

52 Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In *Advances in Cryptology – CRYPTO'09*, pages 619–636, 2009.

53 Brent Waters. A punctured programming approach to adaptively secure functional encryption. In *Advances in Cryptology – CRYPTO'15*, pages 678–697, 2015.

Inferential Privacy Guarantees for Differentially Private Mechanisms

Arpita Ghosh[1] **and Robert Kleinberg**[2]

1 **Cornell University, Ithaca, NY, USA**
arpitaghosh@cornell.edu
2 **Cornell University, Ithaca, NY, USA**
robert.kleinberg@cornell.edu

Abstract

The following is a summary of the paper "Inferential Privacy Guarantees for Differentially Private Mechanisms", presented at the 8^{th} *Innovations in Theoretical Computer Science* conference in January 2017. The full version of the paper can be found on arXiv at the following URL: https://arxiv.org/abs/1603.01508.

1998 ACM Subject Classification K.4.1 Public Policy Issues – Privacy

Keywords and phrases differential privacy, statistical inference, statistical mechanics

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.9

1 Summary of the paper

Differential privacy [3, 4] has become the dominant theoretical framework for quantifying privacy loss, and has begun to make its way into policy and legal frameworks as a potential means for such a quantification. Running a differentially private mechanism on a dataset produces an outcome whose distribution is insensitive to removing one individual's data from the dataset or modifying her data. Consequently, differential privacy guards an individual against the possibility that observers of the mechanism's outcome will make strong inferences about her participation or non-participation (or, contingent on participation, inferences about her data).

Even without an individual's participation in a dataset, probabilistic inferences about her private data may be possible due to its correlation with the data of other individuals present in the dataset. In some cases, but not all, it may be desirable to protect individuals from such inferences. This paper aims to quantify the extent to which differentially private mechanisms guarantee such "inferential privacy protection", as a function of the prior belief (or set of potential priors) held by observers prior to the release of the mechanism's outcome.

We can illustrate the issues at play here using the oft-cited example of a study showing that smoking causes cancer [5]. If the data underlying the study were analyzed in a differentially private manner, readers of the study should not significantly update their beliefs about a given individual's participation in the dataset. However, if the individual were a known smoker, they should significantly revise their beliefs about her probability of developing cancer. Such belief revisions, in this circumstance, are generally not construed as a violation of privacy, because they merely reflect improved knowledge of an aggregate property of the population, not any knowledge specific to the individual. On the other hand, suppose that the hypothetical study focused on the more fine-grained question of how the likelihood of developing smoking-related cancers varies as a function of factors such as age, diet, lifestyle, and family history, and that the supplementary material accompanying the article included a


© Arpita Ghosh and Robert Kleinberg;
licensed under Creative Commons License CC-BY

machine learning model (e.g., a random forest regressor) trained to predict these likelihoods. Even if the input-output behavior of the regressor were not construed as violating privacy, the regressor itself could be a highly complex object (e.g., a collection of more than a thousand decision trees) and its contents might reflect private information in the training data. If the training algorithm were differentially private, would it ensure that participants in the study were protected from "inferential privacy violations"? This paper abstracts away the specifics of the example and asks the basic question: when does a differential privacy guarantee for an algorithm imply an inferential privacy guarantee?

In the paper, we formally define the inferential privacy parameter of a mechanism with respect to a set of prior distributions. Informally, it measures the maximum amount by which an adversary might multiplicatively update his belief in the likelihood that a particular individual's entry in the database assumes a particular value, given that the adversary starts with one of the specified prior distributions and performs a Bayesian update on the mechanism's outcome. An easy application of Bayes' Law confirms the fact (known to many prior authors, e.g. [7]) that when individuals' private data are mutually independent, the differential privacy parameter of a mechanism equals its inferential privacy parameter. On the other hand, we have seen that when the adversary's prior incorporates uncertainty about certain aggregate statistics of the population (e.g., the frequency of cancer among smokers) the inferential privacy parameter of a mechanism may be much greater than its differential privacy parameter. Interpolating between these two extremes, one might guess that differentially private mechanisms are guaranteed to have a relatively small inferential privacy parameter when correlations between individuals are either weak or localized, i.e. when each individual has only a few others with whom her data correlates strongly.

Our work makes this intuition precise and confirms it rigorously. To do so, we identify a surprisingly tight relationship between our questions about privacy and inference and a corresponding set of questions in mathematical physics regarding the magnitude of the change in a Gibbs measure when an external field is applied to the system. While it may initially seem surprising that the two fields are connected in this way, one can see the first intimations of the connection in the discussion about weak, localized correlations at the end of the preceding paragraph—this is none other than the *correlation decay* property that is the hallmark of statistical mechanical systems at high temperature.

Our first main theorem pertains to cases in which the data are binary-valued and the adversary's prior distribution satisfies *positive affiliation*, meaning that conditioning on all but two entries in the database, the remaining two entries can never be negatively correlated. This assumption is satisfied by the priors commonly ascribed to biological data—where heredity and contagion lead to positive correlations among individuals' attributes, but never or rarely lead to negative correlations—and to social data, where positive correlations result from homophily and social contagion. (If one interprets the adversary's prior as a Gibbs measure, the positive affiliation property says that the system is *ferromagnetic*.) Our theorem gives a precise formula for the worst-case inferential privacy parameter of ϵ-differentially private mechanisms in terms of the magnetization of the corresponding ferromagnetic system in an external field of strength $\frac{\epsilon}{2}$. The proof of the theorem shows that the mechanism attaining this worst-case bound is not a contrived mechanism; in fact it is one of the most commonly occurring differentially private mechanisms: adding Laplace noise to the sum of the values in the database! Thus, when data are positively affiliated, any inferential privacy guarantee that one can prove for the Laplace mechanism *automatically* carries over to arbitrary differentially private mechanisms.

Because our theorem provides an exact formula for worst-case inferential privacy in terms of magnetization (and not merely an upper bound) it allows us to translate results about the physics of magnets directly into results about inferential privacy. In particular, existing results about phase transitions in Ising models (e.g., for the infinite d-regular tree [1]) imply that inferential privacy parameters can be surprisingly sensitive to variations in other parameters of the model. For example, if we vary the differential privacy parameter of the Laplace mechanism, starting at $\epsilon = 0$ and increasing from there, the mechanism's inferential privacy parameter increases gradually until ϵ crosses a critical value, at which point it can increase very precipitously, approaching a step discontinuity as the number of individuals tends to infinity. In other words, tiny variations in the differential privacy parameter of a mechanism can potentially lead to enormous variations in inferential privacy, a privacy-theoretic manifestation of the physical phenomenon of phase transitions in spin systems.

Our second main result applies when the data are not binary-valued, or when the prior violates positive affiliation. It shows that the inferential privacy parameter of a mechanism is bounded above by a function of the mechanism's differential privacy parameter and the spectral norm of an *influence matrix* encoding the strength of pairwise correlations between individuals. The theorem again manifests the relationship between inferential privacy and statistical mechanics; its statement and proof constitute an adaptation of the Dobrushin Comparison Theorem [2, 6, 8] from the traditional setting of additive approximation to multiplicative approximation.

Acknowledgements. The authors gratefully acknowledge helpful discussions with Christian Borgs, danah boyd, Jennifer Chayes, Cynthia Dwork, Kobbi Nissim, Adam Smith, Omer Tamuz, and Salil Vadhan.

Much of this research was completed while Robert Kleinberg was a researcher at Microsoft Research New England. Both authors were partially supported by NSF award AF-1512964. Arpita Ghosh was partially supported by NSF award III-1513692.

References

1 Rodney J Baxter. *Exactly solved models in statistical mechanics*. Courier Corporation, 2007.

2 Roland L Dobrushin. Prescribing a system of random variables by conditional distributions. *Theory of Probability & Its Applications*, 15(3):458–486, 1970.

3 Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming (ICALP)*, 2006.

4 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, 2006.

5 Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 2013.

6 H. Föllmer. A covariance estimate for Gibbs measures. *Journal of Functional Analysis*, 46:387–395, 1982.

7 Shiva P Kasiviswanathan and Adam Smith. On the 'semantics' of differential privacy: A Bayesian formulation. *Journal of Privacy and Confidentiality*, 6(1):1, 2014.

8 H Künsch. Decay of correlations under Dobrushin's uniqueness condition and its applications. *Communications in Mathematical Physics*, 84(2):207–222, 1982.

Towards Human Computable Passwords*

Jeremiah Blocki[1], Manuel Blum[2], Anupam Datta[3], and Santosh Vempala[4]

1 **Purdue University, West Lafayette, USA**
jblocki@purdue.edu

2 **Carnegie Mellon University, Pittsburgh, USA**
mblum@cs.cmu.edu

3 **Carnegie Mellon University, Pittsburgh, USA**
danupam@cmu.edu

4 **Georgia Tech, Altanta, USA**
vempala@cc.gatech.edu

Abstract

An interesting challenge for the cryptography community is to design authentication protocols that are so simple that a human can execute them without relying on a fully trusted computer. We propose several candidate authentication protocols for a setting in which the human user can only receive assistance from a semi-trusted computer – a computer that stores information and performs computations correctly but does not provide confidentiality. Our schemes use a semi-trusted computer to store and display public challenges $C_i \in [n]^k$. The human user memorizes a random secret mapping $\sigma : [n] \to \mathbb{Z}_d$ and authenticates by computing responses $f(\sigma(C_i))$ to a sequence of public challenges where $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$ is a function that is easy for the human to evaluate. We prove that any statistical adversary needs to sample $m = \tilde{\Omega}\left(n^{s(f)}\right)$ challenge-response pairs to recover σ, for a security parameter $s(f)$ that depends on two key properties of f. Our lower bound generalizes recent results of Feldman et al. [26] who proved analogous results for the special case $d = 2$. To obtain our results, we apply the general hypercontractivity theorem [45] to lower bound the *statistical dimension* of the distribution over challenge-response pairs induced by f and σ. Our *statistical dimension* lower bounds apply to arbitrary functions $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$ (not just to functions that are easy for a human to evaluate). As an application, we propose a family of human computable password functions f_{k_1,k_2} in which the user needs to perform $2k_1 + 2k_2 + 1$ primitive operations (e.g., adding two digits or remembering a secret value $\sigma(i)$), and we show that $s(f) = \min\{k_1 + 1, (k_2 + 1)/2\}$. For these schemes, we prove that forging passwords is equivalent to recovering the secret mapping. Thus, our human computable password schemes can maintain strong security guarantees even after an adversary has observed the user login to many different accounts.

1998 ACM Subject Classification D.4.6 Authentication

Keywords and phrases Passwords, Cognitive Authentication, Human Computation, Planted Constraint Satisfaction Problem, Statistical Dimension

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.10

* This work was partially supported by the NSF Science and Technology TRUST and the AFOSR MURI on Science of Cybersecurity. This work was completed in part while the first author was at Carnegie Mellon University where the first author partially supported by an NSF Graduate Fellowship.

© Jeremiah Blocki, Manuel Blum, Anupam Datta, and Santosh Vempala;
licensed under Creative Commons License CC-BY

1 Introduction

A typical computer user has many different online accounts which require some form of authentication. While passwords are still the dominant form of authentication, users struggle to remember their passwords. As a result users often adopt insecure password practices (e.g., reuse, weak passwords) [28, 21, 39, 18] or end up having to frequently reset their passwords. Recent large-scale password breaches highlight the importance of this problem [1, 21, 11, 2, 50, 3, 4, 5, 6, 7, 8, 9]. An important research goal is to develop usable and secure password management scheme – a systematic strategy to help users create and remember multiple passwords. Blocki et al. [13] and Blum and Vempala [17] recently proposed password management schemes that maintain some security guarantees after a small constant number of breaches (e.g., an adversary who sees three of the user's passwords still has some uncertainty about the user's remaining passwords).

In this work we focus on the goal of developing human computable password management schemes in which security guarantees are strongly maintained after *many* breaches (e.g., an adversary who sees one-hundred of the user's passwords still has high uncertainty about the user's remaining passwords). In a human computable password management scheme the user reconstructs each of his passwords by *computing* the response to a public challenge.

Our human computable password schemes admittedly require more human effort than the password management schemes of Blocki et al. [13] and Blum and Vempala [17], and, unlike Blocki et al. [13], our scheme requires users to do simple mental arithmetic (e.g., add two single-digit numbers) in their head. However, our proposed schemes are still human usable in the sense that a motivated, security-conscious user would be able to learn to use the scheme and memorize all associated secrets in a few hours. In particular, the human computation in our schemes only involves a few very simple operations (e.g., addition modulo 10) over secret values (digits) that the user has memorized. More specifically, in our candidate human computable password schemes the user learns to compute a simple function $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$,[1] and memorizes a secret mapping $\sigma : [n] \to \mathbb{Z}_d$. The user authenticates by responding to a sequence of single digit challenges, i.e., a challenge-response pair $(C, f(\sigma(C)))$ is a challenge $C \in X_k \subseteq [n]^k$ and the corresponding response.

One of our candidate human computable password schemes involves the function

$$f(x_0, x_1, x_2, x_3, x_4, x_5, \ldots, x_{13}) = x_{13} + x_{12} + x_{(x_{10}+x_{11} \mod 10)} \mod 10 .$$

If the user memorizes a secret mapping σ from n images to digits then each challenge $C = (I_0, \ldots, I_{13})$ would correspond to an ordered subset of 14 of these images and the response to the challenge is $f\big(\sigma(I_0), \ldots, \sigma(I_{13})\big)$. We observe that a human would only need to perform three addition operations modulo 10 to evaluate this function. The user would respond by (1) adding the secret digits associated with challenge images I_{10} and I_{11} to get a secret index $0 \le i \le 9$, (2) finding image I_i, (3) adding the secret digits associated with images I_i, I_{12} and I_{13} to produce the final response. To amplify security the user may respond to $\lambda \ge 1$ single-digit challenges $C_1, \ldots, C_\lambda$ to obtain a λ digit password $f(\sigma(C_1)), \ldots, f(\sigma(C_\lambda))$. We note that the challenge C does not need to be kept secret and thus the images can be arranged on the screen in helpful manner for the human user – see Figure 1 for an example and see Appendix A for more discussion of the user interface.

[1] In our security analysis we consider arbitrary bases d. However, our specific schemes use the base $d = 10$ that is most familiar to human users.

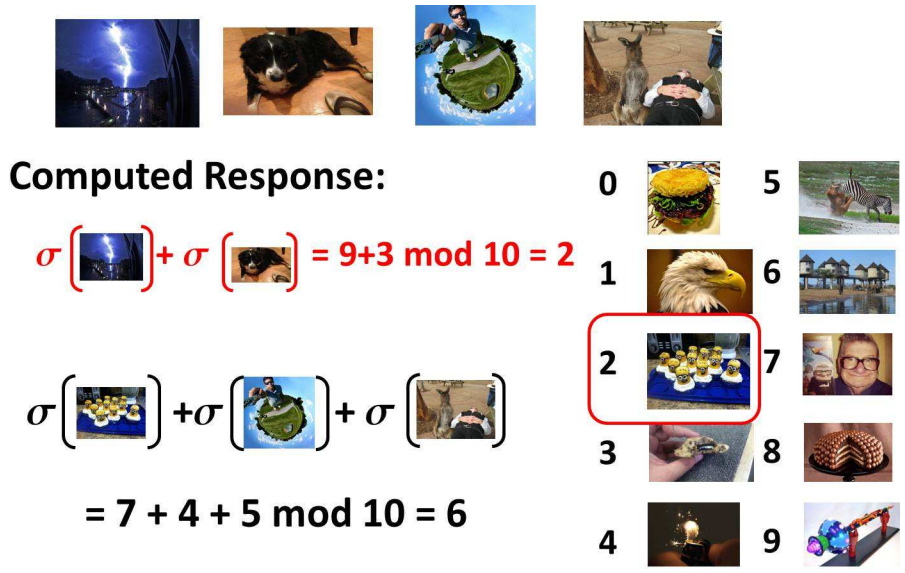


Figure 1 Computing the response $f(\sigma(C)) = 6$ to a single digit challenge C.

We present a natural conjecture which implies that a polynomial time attacker will need to see the responses to $\tilde{\Omega}\left(n^{s(f)}\right)$ random challenges before he can forge the user's passwords (accurately predict the responses to randomly selected challenges)[2]. Here, $s(f)$ is a security parameter that depends on two key properties of the function f (in our above example $s(f) = 3/2$). Furthermore, we provide strong evidence for our conjecture by ruling out a broad class of algorithmic techniques that the adversary might use to attack our scheme.

Following Blocki et al. [13] we consider a setting where a user has two types of memory: *persistent memory* (e.g., a sticky note or a text file on his computer) and *associative memory* (e.g., his own human memory). We assume that persistent memory is reliable and convenient but not private (i.e., an adversary can view all challenges stored in persistent memory, but he cannot tamper with them). In contrast, a user's associative memory is private but lossy – if the user does not rehearse a memory it may be forgotten. Thus, the user can store a password challenge $C \in X_k$ in persistent memory, but the mapping σ must be stored in associative memory (e.g., memorized and rehearsed). We allow the user to receive assistance from a semi-trusted computer. A semi-trusted computer will perform computations accurately (e.g., it can be trusted to show the user the correct challenge), but it will not ensure privacy of its inputs or outputs. This means that a human computable password management scheme should be based on a function f that the user can compute entirely in his head.

Contributions

We provide precise notions of security and usability for a human computable password management scheme (Section 2). We introduce the notion of UF-RCA security (Unforgeability under Random Challenge Attacks). Informally, a human computable password scheme is

[2] We stress that, unlike [13, 17], our security guarantees are not information theoretic. In fact, a computationally unbounded adversary would need to see at most $O(n)$ challenge-response pairs to break the human computable password management scheme.

UF-RCA secure if an adversary cannot forge passwords after seeing many example challenge-response pairs.

We present the design of a candidate family of human computable password management schemes f_{k_1,k_2}, and analyze the usability and security of these schemes (Section 3). Our usability analysis indicates that to compute $f_{k_1,k_2}(\sigma(C))$ the user needs to execute $2k_1+2k_2+1$ simple operations (e.g., addition of single digits modulo 10). The main technical result of this section (Theorem 10) states that our scheme is UF-RCA secure given a plausible conjecture about the hardness of random planted constraint satisfiability problems (*RP-CSP*). Our conjecture is that any polynomial time adversary needs to see at least $m = n^{\min\{r(f)/2, g(f)+1-\epsilon\}}$ challenge-response pairs $(C, f(\sigma(C)))$ to recover the secret mapping σ. Here, $s(f) = \min\{r(f)/2, g(f)+1\}$ is a composite security parameter involving $g(f)$ (how many inputs to f need to be fixed to make f linear?) and $r(f)$ (what is the largest value of r such that the distribution over challenge-response pairs are $(r-1)$-wise independent?). We prove that $g(f_{k_1,k_2}) = k_1$ and $r(f_{k_1,k_2}) = (k_2+1)$.

Next we prove that any statistical adversary needs at least $\tilde{\Omega}\left(n^{r(f)/2}\right)$ challenge-response pairs $(C, f(\sigma(C)))$ to find a secret mapping σ' that is ϵ-correlated with σ (Section 4). This result may be interpreted as strong evidence in favor of the *RP-CSP* hardness assumption as most natural algorithmic techniques have statistical analogues (see discussion in Section 4). While Gaussian Elimination is a notable exception, our composite security parameter accounts for attacks based on Gaussian Elimination – an adversary needs to see $m = \tilde{\Omega}\left(n^{1+g(f)}\right)$ challenge-response pairs to recover σ using Gaussian Elimination. Moving beyond asymptotic analysis we also provide empirical evidence that our human computable password management scheme is hard to crack. In particular, we used a CSP solver to try to recover $\sigma \in \mathbb{Z}_{10}^n$ given m challenge-response pairs using the functions $f_{1,3}$ and $f_{2,2}$. Our CSP solver failed to find the secret mapping $\sigma \in \mathbb{Z}_{10}^{50}$ given $m = 1000$ random challenge-response pairs with both functions $f_{1,3}$ and $f_{2,2}$. Additionally, we constructed public challenges for cryptographers to break our human computable password management schemes under various parameters (e.g., $n = 100$, $m = 1000$).

Our lower bound for statistical adversaries is based on the *statistical dimension* of the distribution over challenge-response pairs induced by f and σ. We stress that our analysis of the statistical dimension applies to arbitrary functions $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$, not just to functions that are easy for humans to compute. Our analysis of the statistical dimension generalizes recent results of Feldman et al. [26] for binary predicates and may be of independent interest. While the analysis is similar at a high level, we stress that our proofs do require some new ideas. Because our function f is not a binary predicate we cannot use the Walsh basis functions to express the Fourier decomposition of f and analyze the statistical dimension of our distribution over challenge-response pairs as Feldman et al. [26] do. Instead, we use a generalized set of Fourier basis functions to take the Fourier decomposition of f, and we apply the general hypercontractivity theorem [45] to obtain our bounds on the statistical dimension.

We complete the proof of Theorem 10 in Section 5 by proving that forging passwords and approximately recovering the secret mapping are equivalent problems for a broad class of human computable password schemes, including our candidate family f_{k_1,k_2}. This result implies that any adversary who can predict the response $f(C)$ to a random challenge C with better accuracy than random guessing can be used as a blackbox to approximately recover the secret mapping.

2 Definitions

2.1 Notation

Given two strings $\alpha_1, \alpha_2 \in \mathbb{Z}_d^n$ we use $H(\alpha_1, \alpha_2) \doteq |\{i \in [n] \mid \alpha_1[i] \neq \alpha_2[i]\}|$ to denote the Hamming distance between them. We will also use $H(\alpha_1) \doteq H(\alpha_1, \vec{0})$ to denote the Hamming weight of α_1. We use $\sigma : [n] \to \mathbb{Z}_d$ to denote a secret random mapping that the user will memorize. We will sometimes abuse notation and think of $\sigma \in \mathbb{Z}_d^n$ as a string which encodes the mapping, and we will use $\sigma \sim \mathbb{Z}_d^n$ to denote a random mapping chosen from $\mathbb{Z}_d^n$ uniformly at random. Given a distribution $\mathcal{D}$ we will use $x \sim \mathcal{D}$ to denote a random sample from this distribution. We also use $x \sim S$ to denote an element chosen uniformly at random from a finite set S.

▶ **Definition 1.** We say that two mappings $\sigma_1, \sigma_2 \in \mathbb{Z}_d^n$ are ϵ-correlated if $\frac{H(\sigma_1,\sigma_2)}{n} \leq \frac{d-1}{d} - \epsilon$, and we say that a mapping $\sigma \in \mathbb{Z}_d^n$ is δ-balanced if $\max_{i \in \{0,\ldots,d-1\}} \left| \frac{H(\sigma,\vec{i})}{n} - \frac{d-1}{d} \right| \leq \delta$.

Note that for a random mapping σ_2 we expect σ_1 and σ_2 to differ at $\mathbb{E}_{\sigma_2 \sim \mathbb{Z}_d^n}[H(\sigma_1, \sigma_2)] = n\left(\frac{d-1}{d}\right)$ locations, and for a random mapping σ and $i \sim \{0, \ldots, d-1\}$ we expect σ to differ from $\vec{i}$ at $\mathbb{E}_{i \sim \mathbb{Z}_d, \sigma \sim \mathbb{Z}_d^n}\left[H\left(\sigma, \vec{i}\right)\right] = n\left(\frac{d-1}{d}\right)$ locations. Thus, with probability $1 - o(1)$ a random mapping σ_2 will not be ϵ-correlated with σ_1, but a random mapping σ will be δ-balanced with probability $1 - o(1)$.

We let $X_k \subseteq [n]^k$ denote the space of ordered clauses of k variables without repetition. We use $C \sim X_k$ to denote a clause C chosen uniformly at random from X_k and we use $\sigma(C) \in \mathbb{Z}_d^k$ to denote the values of the corresponding variables in C. For example, if $d = 10$, $C = (3, 10, 59)$ and $\sigma(i) = (i + 1 \mod 10)$ then $\sigma(C) = (4, 1, 0)$.

We view each clause $C \in X_k$ as a *single-digit challenge*. The user responds to a challenge C by computing $f(\sigma(C))$, where $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$ is a *human computable* function (see discussion below) and $\sigma : [n] \to \mathbb{Z}_d$ is the secret mapping that the user has memorized. For example, if $d = 10$, $C = (3, 10, 59)$, $\sigma(i) = (i + 1 \mod 10)$ and $f(x, y, z) = (x - y + z \mod 10)$ then $f(\sigma(C)) = (4 - 1 + 0 \mod 10) = 3$. A length-$\lambda$ password challenge $\vec{C} = \langle C_1, \ldots, C_\lambda \rangle \in (X_k)^\lambda$ is a sequence of t single digit challenges, and $f(\sigma(\vec{c})) = \langle f(\sigma(C_1)), \ldots, f(\sigma(C_\lambda)) \rangle \in \mathbb{Z}_d^\lambda$ denotes the corresponding response (e.g., a password).

Let's suppose that the user has m accounts $A_1, \ldots, A_m$. In a human computable password management scheme we will generate m length-λ password challenges $\vec{C}_1, \ldots, \vec{C}_m \in (X_k)^\lambda$. These challenges will be stored in persistent memory so they are always accessible to the user as well as the adversary. When our user needs to authenticate to account A_i he will be shown the length-t password challenge $\vec{C}_i = \langle C_1^i, \ldots, C_\lambda^i \rangle$. The user will respond by computing his password $p_i = \langle f(\sigma(C_1^i)), \ldots, f(\sigma(C_\lambda^i)) \rangle \in \mathbb{Z}_d^\lambda$.

2.2 Requirements for a Human Computable Function

In our setting we require that the composite function $f \circ \sigma : X_k \to \mathbb{Z}_d$ is human computable. Informally, we say that a function f is *human-computable* if a human user can evaluate f *quickly* in his head.

▶ **Requirement 2.** *A function f is $\hat{t}$-*human computable *for a human user H if H can reliably evaluate f in his head in $\hat{t}$ seconds.*

We argue that a function f will be *human-computable* whenever there is a fast streaming algorithm [10] to compute f using only very simple primitive operations. A streaming

algorithm is an algorithm for processing a data stream in which the input (e.g., the challenge C) is presented as a sequence of items that can only be examined once. In our context the streaming algorithm must have a very low memory footprint because a typical person can only keep 7 ± 2 'chunks' of information in working memory [41] at any given time. Our streaming algorithm can only involve primitive operations that a person could execute quickly in his head (e.g., adding two digits modulo 10, recalling a value $\sigma(i)$ from memory).

▶ **Definition 3.** Let P be a set of primitive operations. We say that a function f is $(P, \tilde{t}, \hat{m})$-computable if there is a space $\hat{m}$ streaming algorithm $\mathcal{A}$ to compute f using only $\tilde{t}$ operations from P.

In this paper we consider the following primitive operations P: **Add**, **Recall** and **TableLookup**. **Add** : $\mathbb{Z}_{10} \times \mathbb{Z}_{10} \to \mathbb{Z}_{10}$ takes two digits x_1 and x_2 and returns $x_1 + x_2$ mod 10. **Recall** : $[n] \to \mathbb{Z}_{10}$ takes an index i and returns the secret value $\sigma(i)$ that the user has memorized. **TableLookup** : $\mathbb{Z}_{10} \times [n]^{10} \to [n]$ takes a digit x_1 and finds the x_1'th value from a table of 10 indices. We take the view that no human computable function should require users to store intermediate values in long-term memory because the memorization process would necessarily slow down computation. Therefore, we restrict our attention to space $\hat{m}$ streaming algorithms and do not include any primitive operation like **MemorizeValue**.

▶ **Example.** The function $f \circ \sigma(i_1, \ldots, i_5) = \sigma(i_1) + \ldots + \sigma(i_5)$ requires 9 primitive operations (five **Recall** operations and four **Add** operations) and requires space $\hat{m} = 3$ (e.g., we need one slot to store the current total, one slot to store the next value from the data stream and one free slot to execute a primitive operation).

Similar primitive operations have been studied by cognitive physchologists (e.g., [51]). The time γ_H it takes a human user H to execute one primitive operation will typically improve with practice (e.g., [33]). We note that we allow this computation speed constant γ_H to vary from user to user in the same way that two computers might operate at slightly different speeds. We conjecture that, after training, a human user H with a moderate mathematical background will be able to evaluate a $(P, \tilde{t}, 3)$-computable function in $\hat{t} \leq \tilde{t}$ seconds – the first author of this paper found that (after some practice) he could evaluate $(P, 9, 3)$-computable functions in 7.5-seconds ($\gamma_H \leq 1$).

▶ **Conjecture 4.** *Let $P = \{\mathbf{Add}, \mathbf{Recall}, \mathbf{TableLookup}\}$. For each human user H there is a small constant $\gamma_H > 0$ such that any $(P, \tilde{t}, 3)$-computable function f will be $\hat{t}$-human computable for H with $\hat{t} = \gamma_H \tilde{t}$.*

2.3 Password Unforgeability

In the password forgeability game the adversary attempts to guess the user's password for a randomly selected account after he has seen the user's passwords at m/λ other randomly selected accounts. We say that a scheme is UF-RCA (Unforgeability against Random Challenge Attacks) secure if any probabilistic polynomial time adversary fails to guess the user's password with high probability. In the password forgeability game we select the secret mapping $\sigma : [n] \to \mathbb{Z}_d$ uniformly at random along with challenges $C_1, \ldots, C_{m+\lambda} \sim X_k$. The adversary is given the function $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$ and is shown the challenges $C_1, \ldots, C_{m+\lambda}$ as well as the values $f(\sigma(C_i))$ for $i \in \{1, \ldots, m\}$. The game ends when the adversary $\mathcal{A}$ outputs a guess $\langle q_1, \ldots, q_\lambda \rangle \in \mathbb{Z}_d^\lambda$ for the value of $\langle f(\sigma(C_{m+1})), \ldots, f(\sigma(C_{m+\lambda})) \rangle$. We say that the adversary wins if he correctly guesses the responses to all of the challenges $C_{m+1}, \ldots, C_{m+\lambda}$, and we use $\mathbf{Wins}(\mathcal{A}, n, m, \lambda)$ to denote the event that the adversary wins the game (e.g.,

$\forall i \in \{1,\ldots,\lambda\}.q_i = f\left(\sigma\left(C_{m+i}\right)\right)$). We are interested in understanding how many example single digit challenge-response pairs the adversary needs to see before he can start breaking the user's passwords.

▶ **Definition 5** (Security)**.** We say that a function $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$ is $\mathbf{UF-RCA}\left(n, m, \lambda, \delta\right)$ – *secure* if for every probabilistic polynomial time (in n, m) adversary $\mathcal{A}$ we have

$$\Pr\left[\mathbf{Wins}\left(\mathcal{A}, n, m, \lambda\right)\right] \leq \delta \ ,$$

where the randomness is taken over the selection of the secret mapping $\sigma \sim \mathbb{Z}_d^n$, the challenges $C_1, \ldots, C_{m+\lambda}$ as well as the adversary's coins.

Discussion

Our security model is different from the security model of Blocki et al. [13] in which the adversary gets to adaptively select which accounts to compromise and which account to attack. While our security model may seem weaker at first glance because the adversary does not get to select which account to compromise/attack, we observe that the password management schemes of Blocki et al. [13] are only secure against one to three adaptive breaches. By contrast, our goal is to design human computable password schemes that satisfy $\mathbf{UF-RCA}$ security for large values of m (e.g. 1000), which means that it is reasonable to believe that the user has at most m/λ password protected accounts. If the user has at most m/λ accounts then union bounds imply that an adaptive adversary – who gets to compromise all but one account – will not be able to forge the password at any remaining account with probability greater than $m\delta/\lambda$ (typically, $m \ll \lambda/\delta$)[3].

2.4 Security Parameters of f

Given a function $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$ we define the function $Q^f : \mathbb{Z}_d^{k+1} \to \{\pm 1\}$ s.t. $Q^f\left(x, i\right) = 1$ if $f(x) = i$; otherwise $Q^f\left(x, i\right) = -1$. We use Q_σ^f to define a distribution over $X_k \times \mathbb{Z}_d$ (challenge-response pairs) as follows: $\Pr_{Q_\sigma^f}\left[C, i\right] \doteq \frac{Q^f(\sigma(C), i)+1}{2|X_k|}$. Intuitively, Q_σ^f is the uniform distribution over challenge response pairs (C, j) s.t. $f\left(\sigma\left(C\right)\right) = j$. We also use $Q^{f,j} : \mathbb{Z}_d^k \to \{\pm 1\}$ $\left(Q^{f,j}\left(x\right) = Q^f\left(x, j\right)\right)$ to define a distribution over X_k. $\Pr_{Q_\sigma^{f,j}}[C] = \frac{Q^{f,j}(f(\sigma(C)))+1}{2|\{C' \in X_k : f(\sigma(C'))=j|} = \Pr_{Q_\sigma^f}\left[(C, i) \mid i = j\right]$. We write the Fourier decomposition of a function $Q : \mathbb{Z}_d^k \to \{\pm 1\}$ as follows

$$Q(x) = \sum_{\alpha \in \mathbb{Z}_d^k} \hat{Q}_\alpha \cdot \chi_\alpha\left(x\right) \ \text{, where the basis functions are } \ \chi_\alpha\left(x\right) \doteq \exp\left(\frac{-2\pi\sqrt{-1}\left(x \cdot \alpha\right)}{d}\right) .$$

We say that a function Q has degree ℓ if $\ell = \max\left\{H\left(\alpha\right) \;\middle|\; \alpha \in \mathbb{Z}_d^k \wedge \hat{Q}_\alpha \neq 0\right\}$ – equivalently if $Q(x) = \sum_i Q_i(x)$ can be expressed as a sum of functions where each function $Q_i : \mathbb{Z}_d^k \to \mathbb{R}$ depends on at most ℓ variables.

▶ **Definition 6.** We use $r(Q) \doteq \min\left\{H\left(\alpha\right) \;\middle|\; \exists \alpha \in \mathbb{Z}_d^k . \hat{Q}_\alpha \neq 0 \wedge \alpha \neq \vec{0}\right\}$ to denote the distributional complexity of Q, and we use $r(f) = \min\left\{r\left(Q^{f,j}\right) \mid j \in \mathbb{Z}_d\right\}$ to denote the distributional complexity of f. We use $g(f) \doteq$

$$\min\left\{\ell \in \mathbb{N} \cup \{0\} \;\middle|\; \exists \alpha \in \mathbb{Z}_d^\ell, S \subseteq [k], \hat{d} \in \mathbb{Z}_d . \text{s.t } |S| = \ell \ \&\ f_{|S,\alpha} \text{ is a linear function mod } \hat{d}\right\} \ ,$$

[3] We assume in our analysis that the adversary does not get to pick the challenges C that the user will solve.

to denote the minimum number of variables that must be fixed to make f a linear function. Here, $f_{|S,\alpha} : \mathbb{Z}_d^{k-\ell} \to \mathbb{Z}_d$ denotes the function f after fixing the variables at the indices specified by S to α. Finally, we use $s(f) \doteq \min\{r(f)/2, g(f)+1\}$ as our composite security measure.

We conjecture that a polynomial time adversary will need to see $m = n^{s(f)}$ challenge-response pairs before he can approximately recover the secret mapping σ. We call this conjecture about the hardness of random planted constraint satisfiability problems RP-CSP (Conjecture 7). In support of RP-CSP we prove that any statistical algorithm needs to see at least $m = \tilde{\Omega}\left(n^{r(f)/2}\right)$ challenge response pairs to (approximately) recover the secret mapping σ and we observe that a polynomial time adversary would need to see $m = O\left(n^{g(f)+1}\right)$ challenge-response pairs to recover σ using Gaussian Elimination. In Section 5 we show that the human computable password scheme will be UF-RCA secure provided that RP-CSP holds and that f satisfies a few moderate properties (e.g., the output of f is evenly distributed).

▶ **Conjecture 7** (RP-CSP). *For every probabilistic polynomial time adversary $\mathcal{A}$ and every $\epsilon, \epsilon' > 0$ there is an integer N s.t. for all $n > N$, $m \leq n^{\min\left\{r(f)/2, g(f)+1-\epsilon'\right\}}$ we have* $\Pr\left[\textbf{Success}\left(\mathcal{A}, n, m, \epsilon\right)\right] \leq \mu(n)$, *where* $\textbf{Success}\left(\mathcal{A}, n, m, \epsilon\right)$ *denotes the event that $\mathcal{A}$ finds a mapping σ' that is ϵ-correlated with σ given m randomly selected challenge response pairs* $\left(C_1, f\left(\sigma\left(C_1\right)\right)\right), \ldots, \left(C_m, f\left(\sigma\left(C_m\right)\right)\right)$ *and $\mu(n)$ is a negligible function. The probability is over the selection of the random mapping σ, the challenges $C_1, \ldots, C_m$ and the random coins of the adversary.*

3 Candidate Secure Human Computable Functions

In this section we present a family of candidate human computable functions. We consider the usability of these human computable password schemes in Section 3.1, and we analyze the security of our schemes in Section 3.2.

We first introduce our family of candidate human computable functions (for all of our candidate human computable functions $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$ we fix $d = 10$ because most humans are used to performing arithmetic operations on digits). Given integers $k_1 > 0$ and $k_2 > 0$ we define the function $f_{k_1,k_2} : \mathbb{Z}_{10}^{10+k_1+k_2}$ as follows

$$f_{k_1,k_2}\left(x_0, \ldots, x_{9+k_1+k_2}\right) = x_j + \sum_{i=10+k_1}^{9+k_1+k_2} x_i \mod 10, \quad \text{where} \quad j = \left(\sum_{i=10}^{9+k_1} x_i\right) \mod 10\ .$$

Authentication Process

We briefly overview the authentication process – see Algorithms 2 and 3 in Appendix A for a more formal presentation of the authentication process. We assume that the mapping $\sigma : \{1, ..., n\} \to \mathbb{Z}_{10}$ is generated by the user's local computer in secret. The user may be shown mnemonic helpers (see discussion below) to help memorize σ, but these mnemonic helpers are discarded immediately afterward. After the user has memorized σ he can create a password pw_i for an account A_i as follows: the user's local computer generates λ random single-digit challenges $C_1^i, \ldots, C_\lambda^i \in X_\lambda$ and the user computes $pw_i = f\left(\sigma\left(C_1\right)\right), \ldots, f\left(\sigma\left(C_\lambda\right)\right)$. The authentication server for account A_i stores the cryptographic hash of pw_i, while the challenges $C_1^i, \ldots, C_\lambda^i \in X_\lambda$ are stored in public memory (e.g., on the user's local computer), which means that they can be viewed by the adversary as well as the legitimate user. To authenticate the user retrieves the public challenges $C_1^i, \ldots, C_\lambda^i$ for account A_i and computes pw_i. The server for A_i verifies that the cryptographic hash of pw_i matches its records. To protect users from offline attacks in the event of a server breach, the password pw_i should be stored using a slow cryptographic hash function $\mathbf{H}$ like BCRYPT [47].

3.1 Usability

In our discussion of usability we focus on the time it would take a human user to compute a password once he has memorized the secret mapping σ. Other important considerations include the challenge of memorizing and rehearsing the secret mapping σ to ensure that the user remembers the secret mapping σ over time.

3.1.1 Computation Time

Given a challenge $C = (c_0, \ldots, c_{9+k_1+k_2}) \in X_{10+k_1+k_2}$ we can compute $f_{k_1,k_2}(\sigma(C))$ we compute $j = \sum_{i=10}^{9+k_1} \sigma(c_i) \mod 10$ using $k_1 - 1$ **Add** operations and k_1 **Recall** operations. We then execute **TableLookup**$(j, c_0, \ldots, c_9)$ to obtain c_j. Now we need k_2 **Add** operations and $k_2 + 1$ **Recall** operations to compute the final response $\sigma(c_j) + \sigma(c_{10+k_1}) + \ldots + \sigma(c_{9+k_1+k_2})$.

▶ **Fact 8.** *Let* $P = \{\textbf{Add}, \textbf{Recall}, \textbf{TableLookup}\}$ *then* $f_{k_1,k_2} \circ \sigma$ *is* $(P, 2k_1 + 2k_2 + 1, 3)$*-computable.*

Fact 8 and Conjecture 4 would imply that $f_{1,3}$ and $f_{2,2}$ are $\hat{t}$-human computable with $\hat{t} = 9$ seconds for humans H with computation constant $\gamma_H \leq 1$. The functions $f_{1,3}$ and $f_{2,2}$ were both $\hat{t}$-human computable with $\hat{t} = 7.5$ seconds for the main author of this paper. While the value of γ_H might be larger for many human users who are less comfortable with mental arithmentic, we note we may have $\gamma_H \ll 1$ for many human users after training (e.g., see `https://youtu.be/_-2L6ZxFacg` for a particularly impressive demonstration of mental arithmetic by young children.).

3.1.2 Memorizing and Rehearsing σ

Memorizing the secret mapping might be the most difficult part of our schemes. In practice, we envision that the user memorizes a mapping from n objects (e.g., images) to digits. For example, if $n = 26$ and $d = 10$ then the user might memorize a random mapping from characters to digits. The first author of this paper was able to memorize a mapping from $n = 100$ images to digits in about 2 hours. We conjecture that the process could be further expedited using mnemonic helpers – see discussion in the appendix.

After the user memorizes σ he may need to rehearse parts of the mapping periodically to ensure that he does not forget it. One of the benefits of our human computable password schemes is that the user will get lots of practice rehearsing the secret mapping each time he computes a password. In fact users who authenticate frequently enough will not need to spend any extra time rehearsing the secret mapping as they will get sufficient natural practice to remember σ.

3.2 Security Analysis

Claim 9 demonstrates that $s(f_{k_1,k_2}) = \min\{(k_2+1)/2, k_1+1\}$. Intuitively, the security of our human computable password management scheme will increase with k_1 and k_2. However, the work that the user needs to do to respond to each single-digit challenge is proportional to $2k_1 + 2k_2 + 1$ (See Fact 8).

▶ **Claim 9.** *Let* $0 \leq k_1$ *and* $k_2 > 0$ *be given and let* $f = f_{k_1,k_2}$ *we have* $g(f) = \min\{k_1, 10\}$, $r(f) = k_2 + 1$ *and* $s(f) = \min\left\{\frac{k_2+1}{2}, k_1 + 1, 11\right\}$.

An intuitive way to see that $r(f_{k_1,k_2}) > k_2$ is to observe that we cannot bias the output of f_{k_1,k_2} by fixing k_2 variables. Fix the value of *any* k_2 variables and draw the values for the other $k_1 + 10$ variables uniformly at random from $\mathbb{Z}_{10}$. One of the $k_2 + 1$ variables in the sum $x_j + \sum_{i=10+k_1}^{9+k_1+k_2} x_i \mod 10$ will not be fixed. Thus, the probability that the final output of $f_{k_1,k_2}(x_0, \ldots, x_{9+k_1+k_2})$ will be r is exactly $1/10$ for each digit $r \in \mathbb{Z}_{10}$. Similarly, an intuitive way to see that $r(f_{k_1,k_2}) \leq k_2 + 1$ is to observe that we can bias the value of $f_{k_1,k_2}(x_0, \ldots, x_{9+k_1+k_2})$ by fixing the value of $k_2 + 1$ variables. In particular if we fix the variables $x_0, x_{10+k_1}, \ldots, x_{9+k_1+k_2}$ so that $0 = x_0 + \sum_{i=10+k_1}^{9+k_1+k_2} x_i \mod 10$ then the output of $f_{k_1,k_2}(x_0, \ldots, x_{9+k_1+k_2})$ is more likely to be 0 than any other digit. The full proof of Claim 9 can be found in Appendix F.

Theorem 10 states that our human computable password management scheme is UF-RCA secure as long as RP-CSP (Conjecture 7) holds. In Section 4 we provide strong evidence in support of RP-CSP. In particular, no statistical algorithm can approximately recover the secret mapping given $m = \tilde{O}\left(n^{r(f)/2}\right)$ challenge-response pairs. To prove Theorem 10 we need to show that an adversary that breaks UF-RCA security for f_{k_1,k_2} can be used to approximately recover the secret mapping σ. We prove a more general result in Section 5.

▶ **Theorem 10.** *Let $\epsilon, \epsilon' > 0, \lambda \geq 1$ be given. Under the RP-CSP conjecture (Conjecture 7) the human computable password scheme defined by f_{k_1,k_2} is $\mathbf{UF-RCA}(n, m, \lambda, \delta)$ − secure for any $m \leq n^{\min\{(k_2+1)/2, k_1+1-\epsilon'\}} - \lambda$ and $\delta > \left(\frac{1}{10} + \epsilon\right)^{\lambda}$.*

▶ Remark. In the Appendix we demonstrate that our security bounds are asymptotically tight. In particular, there is a statistical algorithm to break our human computable password schemes (f_{k_1,k_2}) which requires $m = \tilde{O}\left(n^{(k_2+1)/2}\right)$ to 1-MSTAT to recover σ (See Theorem 27 in Section G). We also demonstrate that there is a attack based on Gaussian Elimination that uses $m = \tilde{O}\left(n^{k_1+1}\right)$ challenge-response pairs to recover σ.

3.2.1 Exact Security Bounds

We used the Constraint Satisfaction Problem solver from the Microsoft Solver Foundations library to attack our human computable password scheme[4]. In each instance we generated a random mapping $\sigma : [n] \to \mathbb{Z}_{10}$ and m random challenge response pairs $(C, f(\sigma(C)))$ using the functions $f_{2,2}$ and $f_{1,3}$. We gave the CSP solver 2.5 days to find σ on a computer with a 2.83 GHz Intel Core2 Quad CPU and 4 GB of RAM. The full results of our experiments are in Appendix C. Briefly, the solver failed to find the random mapping in the following instances with $f = f_{2,2}$ and $f = f_{1,3}$: (1) $n = 30$ and $m = 100$, (2) $n = 50$ and $m = 1,000$ and (3) $n = 100$ and $m = 10,000$.

▶ Remark. While the theoretical security parameter for $f_{1,3}$ ($s(f_{1,3}) = 2$) is slightly better than the security parameter for $f_{2,2}$ ($s(f_{2,2}) = 1.5$), we conjecture that $f_{2,2}$ may be more secure for small values of n (e.g., $n \leq 100$) because it is less vulnerable to attacks based on Gaussian Elimination. In particular, there is a polynomial time attack on $f_{1,3}$ based on Gaussian Elimination that requires at most n^2 examples to recover σ, while the same attack would require n^3 examples with $f_{2,2}$. Our CSP solver was not able to crack $\sigma \in \mathbb{Z}_{10}^{100}$ given $10,000 = 100^2$ challenge response pairs with $f_{2,2}$.

[4] Thanks to David Wagner for suggesting the use of SAT solvers.

Human Computable Password Challenge.

We are challenging the security and cryptography community to break our human computable password scheme for instances that our CSP solver failed to crack (see Appendix B for more details about the challenge). Briefly, for each challenge we selected a random secret mapping $\sigma \in \mathbb{Z}_{10}^n$, and published (1) m single digit challenge-response pairs $(C_1, f(\sigma(C_1))),\ldots,$ $(C_m, f(\sigma(C_m)))$, where each clause C_i is chosen uniformly at random from X_k, and (2) 20 length–$\lambda = 10$ password challenges $\vec{C}_1, \ldots, \vec{C}_{20} \in (X_k)^{10}$. The goal of each challenge is to correctly guess one of the secret passwords $p_i = f\left(\sigma\left(\vec{C}_i\right)\right)$ for some $i \in [20]$. The challenges can be found at `http://www.cs.cmu.edu/~jblocki/HumanComputablePasswordsChallenge/challenge.htm`. There is a \$20 prize associated with each individual challenge (total: \$360). We remark that these challenges remain unsolved even after they were presented during the rump sessions at a cryptography conference and a security conference[12].

4 Statistical Adversaries and Lower Bounds

Our main technical result (Theorem 14) is a lower bound on the number of single digit challenge-response pairs that a statistical algorithm needs to see to (approximately) recover the secret mapping σ. Our results are quite general and may be of independent interest. Given *any* function $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$ we prove that *any* statistical algorithm needs $\tilde{\Omega}\left(n^{r(f)/2}\right)$ examples before it can find a secret mapping $\sigma' \in \mathbb{Z}_d^n$ such that σ' is ϵ-correlated with σ. We first introduce statistical algorithms in Section 4.1 before stating our main lower bound for statistical algorithms in Section 4.2. We also provide a high level overview of our proof in Section 4.2.

4.1 Statistical Algorithms

A statistical algorithm is an algorithm that solves a distributional search problem $\mathcal{Z}$. In our case the distributional search problem $\mathcal{Z}_{\epsilon,f}$ is to find a mapping τ that is ϵ-correlated with the secret mapping σ given access to m samples from Q_σ^f – the distribution over challenge response pairs induced by σ and f. A statistical algorithm can access the input distribution Q_σ^f by querying the 1-MSTAT oracle or by querying the VSTAT oracle (Definition 11).

▶ **Definition 11.** [26] [1-MSTAT(L) oracle and VSTAT oracle] Let D be the input distribution over the domain X. Given any function $h : X \to \{0, 1, \ldots, L-1\}$, 1-MSTAT($L$) takes a random sample x from D and returns $h(x)$. For an integer parameter $T > 0$ and any query function $h : X \to \{0, 1\}$, VSTAT (T) returns a value $v \in [p - \tau, p + \tau]$ where $p = \mathbb{E}_{x\sim D}[h(x)]$ and $\tau = \max\left\{\frac{1}{T}, \sqrt{\frac{p(1-p)}{T}}\right\}$.

In our context the domain $X = X_k \times \mathbb{Z}_d$ is the set of all challenge response pairs and the distribution $D = Q_\sigma^f$ is the uniform distribution over challenge-response pairs induced by σ and f. Feldman et al. [26] used the notion of statistical dimension (Definition 12) to lower bound the number of oracle queries necessary to solve a distributional search problem (Theorem 13). Before we can present the definition of statistical dimension we need to introduce the *discrimination norm*. Intuitively, if the discrimination norm is small then a statistical algorithm will (whp) not be able to distinguish between honest samples $(C, f(\sigma(C))$ and samples from reference distribution T over $X_k \times \mathbb{Z}_d$ which is completely

independent of σ [5]. We define our reference distribution as follows:

$$\Pr_T[(C,i)] = \frac{\Pr_{x\sim\mathbb{Z}_d^k}[f(x)=i]}{|X_k|} \ .$$

Now given a set $\mathcal{D}' \subseteq \mathbb{Z}_d^n$ of secret mappings the discrimination norm of $\mathcal{D}'$ is denoted by $\kappa_2(\mathcal{D}')$ and defined as follows:

$$\kappa_2(\mathcal{D}') \doteq \max_{h,\|h\|=1} \left\{\mathbb{E}_{\sigma\sim\mathcal{D}'}\left[\left|\Delta\left(h,\sigma\right)\right|\right]\right\} \ ,$$

where $h : X_k \times \mathbb{Z}_d \to \mathbb{R}$, $\|h\| \doteq \sqrt{\mathbb{E}_{(C,i)\sim X_k\times\mathbb{Z}_d}\left[h^2\left(C,i\right)\right]}$ and

$$\Delta\left(h,\sigma\right) \doteq \mathbb{E}_{C\sim X_k}\left[h\left(C, f\left(\sigma\left(C\right)\right)\right)\right] - \mathbb{E}_{(C,i)\sim T}\left[h\left(C,i\right)\right] \ .$$

▶ **Definition 12.** [26][6]. For $\kappa > 0$, $\eta > 0$, $\epsilon > 0$, let d' be the largest integer such that for any mapping $\sigma \in \mathbb{Z}_d^n$ the set $\mathcal{D}_\sigma = \mathbb{Z}_d^n \setminus \{\sigma' \in \mathbb{Z}_d^n \mid \sigma' \text{ is } \epsilon\text{-correlated with } \sigma\}$ has size at least $(1-\eta)\cdot|\mathbb{Z}_d^n|$ and for any subset $\mathcal{D}' \subseteq \mathcal{D}_\sigma$ where $|\mathcal{D}'| \geq |\mathcal{D}_\sigma|/d'$, we have $\kappa_2(\mathcal{D}') \leq \kappa$. The **statistical dimension** with discrimination norm κ and error parameter η is d' and denoted by $\mathrm{SDN}(\mathcal{Z}_{\epsilon,f}, \kappa, \eta)$.

Feldman et al. [26] proved the following lower bound on the number of 1-MSTAT and VSTAT queries needed to solve a distributional search problem. Intuitively, Theorem 13 implies that many queries are needed to solve a distributional search problem with high statistical dimension. In Section 4.2 we argue that the statistical dimension our distributional search problem (finding σ' that is ϵ-correlated with the secret mapping σ given m samples from the distribution Q_σ^f) is high.

▶ **Theorem 13.** *[26, Theorems 10 and 12] For $\kappa > 0$ and $\eta \in (0,1)$ let $d' = \mathrm{SDN}(\mathcal{Z}_{\epsilon,f}, \kappa, \eta)$ be the statistical dimension of the distributional search problem $\mathcal{Z}_{\epsilon,f}$. Any randomized statistical algorithm that, given access to a VSTAT$\left(\frac{1}{3\kappa^2}\right)$ oracle (resp. 1-MSTAT(L)) for the distribution Q_σ^f for a secret mapping σ chosen randomly and uniformly from $\mathbb{Z}_d^n$, succeeds in finding a mapping $\tau \in \mathbb{Z}_d^n$ that is ϵ-correlated with σ with probability $\Lambda > \eta$ over the choice of distribution and internal randomness requires at least $\frac{\Lambda-\eta}{1-\eta}d'$ (resp. $\Omega\left(\frac{1}{L}\min\left\{\frac{d'(\Lambda-\eta)}{1-\eta}, \frac{(\Lambda-\eta)^2}{\kappa^2}\right\}\right)$) calls to the oracle.*

As Feldman et al. [26] observe, almost all known algorithmic techniques can be modeled within the statistical query framework. In particular, techniques like Expectation Maximization[25], local search, MCMC optimization[30], first and second order methods for convex optimization, PCA, ICA, k-means can be modeled as a statistical algorithm even with $L = 2$ – see [16] and [22] for proofs. One issue is that a statistical simulation might need polynomially more samples. However, for $L > 2$ we can think of our queries to 1-MSTAT(L) as evaluating L disjoint functions on a random sample. Indeed, Feldman et al. [26] demonstrate that there is a statistical algorithm for binary planted satisfiability problems using $\tilde{O}\left(n^{r(f)/2}\right)$ calls to 1-MSTAT $\left(n^{\lceil r(f)/2\rceil}\right)$.

---

[5] Observe that this implies that a statistical algorithm cannot find the secret σ. In particular, because the distribution T is independent of the secret mapping σ samples from T will not leak any information about σ.

[6] For the sake of simplicity we define the discrimination norm and the statistical dimension using our particular distributional search problem $\mathcal{Z}_{\epsilon,f}$. Our definition is equivalent to the definition in [26] once we fix the reference distribution T.

▶ Remark. We can also use the statistical dimension to lower bound the number of queries that an algorithm would need to make to other types of statistical oracles to solve a distributional search problem. For example, we could also consider an oracle MVSTAT(L,T) that takes a query $h : X \to \{0,\ldots,L-1\}$ and a set $\mathcal{S}$ of subsets of $\{0,\ldots,L-1\}$ and returns a vector $v \in \mathbb{R}^L$ s.t for every $Z \in \mathcal{S}$

$$\left|\sum_{i\in Z} v[i] - p_Z\right| \leq \max\left\{\frac{1}{T}, \sqrt{\frac{p_Z\left(1-p_Z\right)}{T}}\right\},$$

where $p_Z = \Pr_{x\sim D}\left[h(x) \in Z\right]$ and the cost of the query is $|\mathcal{S}|$. Feldman et al. [26, Theorem 7] proved lower bounds similar to Theorem 13 for the MVSTAT oracle. In this paper we focus on the 1-MSTAT and VSTAT oracles for simplicity of presentation.

4.2 Statistical Dimension Lower Bounds

We are now ready to state our main technical result[7].

▶ **Theorem 14.** *Let $\sigma \in \mathbb{Z}_d^n$ denote a secret mapping chosen uniformly at random, let Q_σ^f be the distribution over $X_k \times \mathbb{Z}_d$ induced by a function $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$ with distributional complexity $r = r(f)$. Any randomized statistical algorithm that finds an assignment τ such that τ is $\left(\sqrt{\frac{-2\ln(\eta/2)}{n}}\right)$-correlated with σ with probability at least $\Lambda > \eta$ over the choice of σ and the internal randomness of the algorithm needs at least m calls to the 1-MSTAT(L) oracle (resp. VSTAT$\left(\frac{n^r}{2(\log n)^{2r}}\right)$ oracle) with $m \cdot L \geq c_1\left(\frac{n}{\log n}\right)^r$ (resp. $m \geq n^{c_1 \log n}$) for a constant $c_1 = \Omega_{k,1/(\Lambda-\eta)}(1)$ which depends only on the values k and $\Lambda-\eta$. In particular if we set $L = \left(\frac{n}{\log n}\right)^{r/2}$ then our algorithms needs at least $m \geq c_1\left(\frac{n}{\log n}\right)^{r/2}$ calls to 1-MSTAT(L).*

The proof of Theorem 14 follows from Theorems 16 and 13. Theorems 14 and 16 generalize results of Feldman et al. [26] which only apply for binary predicates $f : \{0,1\}^k \to \{0,1\}$. An interested reader can find our proofs in Appendix D. At a high level our proof proceeds as follows: Given any function $h : X_k \times \mathbb{Z}_d \to \mathbb{R}$ we show that $\Delta\left(\sigma,h\right)$ can be expressed in the following form: $\Delta\left(\sigma,h\right) = \sum_{\ell=r(f)}^{k} \frac{1}{|X_\ell|} b_\ell\left(\sigma\right)$, where $|X_\ell| = \Theta\left(n^\ell\right)$ and each function b_ℓ has degree ℓ (Lemma 21). We then use the general hypercontractivity theorem [45, Theorem 10.23] to obtain the following concentration bound.

▶ **Lemma 15.** *Let $b : \mathbb{Z}_d^n \to \mathbb{R}$ be any function with degree at most ℓ, and let $\mathcal{D}' \subseteq \mathbb{Z}_d^n$ be a set of assignments for which $d' = d^n/\left|\mathcal{D}'\right| \geq e^\ell$. Then $\mathbb{E}_{\sigma\sim\mathcal{D}'}\left[\left|b\left(\sigma\right)\right|\right] \leq 2(\ln d'/c_0)^{\ell/2}\|b\|_2$, where $c_0 = \ell\left(\frac{1}{2ed}\right)$ and $\|b\|_2 = \sqrt{\mathbb{E}_{x\sim\mathbb{Z}_d^n}\left[b\left(x\right)^2\right]}$.*

We then use Lemma 15 to bound $\mathbf{E}_{\sigma\sim\mathcal{D}'}\left[\Delta\left(\sigma,h\right)\right]$ for any set $\mathcal{D}' \subseteq \mathbb{Z}_d^n$ such that $\left|\mathcal{D}'\right| = \left|\mathbb{Z}_d^k\right|/d'$ (Lemma 25). This leads to the following bound on $\kappa_2(\mathcal{D}') = O_k\left(\left(\ln d'/n\right)^{r(f)/2}\right)$.

[7] We remark that for our particular family of human computable functions f_{k_1,k_2} we could get a theorem similar to Theorem 14 by selecting $\sigma \sim \{0,5\}^n$ and appealing directly to results of Feldman et al. [26]. However, this theorem would be weaker than Theorem 14 as it would only imply that a statistical algorithm cannot find an assignment σ' that is $\frac{1}{2} - \frac{1}{10} + \epsilon$-correlated with σ for $\epsilon > 0$. In contrast, our theorem implies that we cannot find σ' that is ϵ-correlated.

▶ **Theorem 16.** *There exists a constant $c_Q > 0$ such that for any $\epsilon > 1/\sqrt{n}$ and $q \geq n$ we have*

$$\mathrm{SDN}\left(\mathcal{Z}_{\epsilon,f}, \frac{c_Q\,(\log q)^{r/2}}{n^{r/2}}, 2e^{-n\cdot\epsilon^2/2}\right) \geq q\ ,$$

where $r = r(f)$ is the distributional complexity of f.

Discussion

We view Theorem 14 as strong evidence for RP-CSP (Conjecture 7) because almost all known algorithmic techniques can be modeled within the statistical query framework[16, 22]. Thus, Theorem 14 rules out most known attacks that an adversary might mount. It also implies that many popular heuristic based SAT solvers (e.g., DPLL[24]) will not be able to recover σ in polynomial time. While Theorem 14 does not rule our attacks based on Gaussian Elimination we consider this class of attacks separately. We need $m = \tilde{O}\left(n^{g(f)+1}\right)$ examples to extract $O(n)$ linear constraints and solve for σ (see Appendix G.2). However, our composite security parameter $s(f) \geq g(f) + 1$ accounts for attacks based on Gaussian Elimination.

5 Security Analysis

In the last section we presented evidence in support of RP-CSP (Conjecture 7) by showing that any statistical adversary needs $m = \tilde{\Omega}\left(n^{r(f)/2}\right)$ examples to (approximately) recover σ. However, RP-CSP only says that it is hard to (approximately) recover the secret mapping σ, not that it is hard to forge passwords. As an example consider the following NP-hard problem from learning theory: find a 2-term DNF that is consistent with the labels in a given dataset. Just because 2-DNF is hard to learn in the *proper* learning model does not mean that it is NP-hard to learn a good classifier for 2-DNF. Indeed, if we allow our learning algorithm to output a linear classifier instead of a 2-term DNF then 2-DNF is easy to learn [37]. Could an adversary win our password security game without properly learning the secret mapping?

Theorem 18, our main result in this section, implies that the answer is no. Informally, Theorem 18 states that any adversary that breaks UF-RCA security of our human computable password scheme f_{k_1,k_2} can also (approximately) recover the secret mapping σ. This implies that our human computable password scheme is UF-RCA secure as long as RP-CSP holds. Of course, for some functions it is very easy to predict challenge-response pairs without learning σ. For example, if f is the constant function – or any function highly correlated with the constant function – then it is easy to predict the value of $f\left(\sigma\left(C\right)\right)$. However, any function that is highly correlated with a constant function is a poor choice for a human computable passwords scheme. We argue that any adversary that can win the password game can be converted into an adversary that properly learns σ provided that the output of function f is evenly distributed (Definition 17).

▶ **Definition 17.** We say that the output of a function $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$ is *evenly distributed* if there exists a function $g : \mathbf{Z}_d^{k-1} \to \mathbb{Z}_d$ such that $f\left(x_1, \ldots, x_k\right) = g\left(x_1, \ldots, x_{k-1}\right) + x_k \mod d$.

Clearly, our family f_{k_1,k_2} has evenly distributed output. To see this we simply set $g = f_{k_1,k_2-1}$. We are now ready to state our main result from this section.

▶ **Theorem 18.** *Suppose that f has evenly distributed output, but that f is not* **UF − RCA** (n, m, λ, δ) *− secure for* $\delta > \left(\frac{1}{d} + \epsilon\right)^{\lambda}$. *Then there is a probabilistic polynomial time algorithm (in n, m, λ and $1/\epsilon$) that extracts a string $\sigma' \in \mathbb{Z}_d^n$ that is $\epsilon/8$-correlated with σ with probability at least $\frac{\epsilon^3}{(8d)^2}$ after seeing $m + \lambda$ example challenge response pairs.*

The proof of Theorem 18 is in Appendix E. We overview the proof here. The proof of Theorem 18 uses Theorem 19 as a subroutine. Theorem 19 shows that we can, with reasonable probability, find a mapping σ' that is correlated with σ given predictions of $f(\sigma(C))$ for each clause as long as the probability that each prediction is accurate is slightly better than a random guess (e.g., $\frac{1}{d} + \delta$). The proof of Theorem 19 is in Appendix E.

▶ **Theorem 19.** *Let f be a function with evenly distributed output (Definition 17), let $\sigma \sim \mathbb{Z}_d^n$ denote the secret mapping, let $\epsilon > 0$ be any constant and suppose that for every $C \in X_k$ we are given labels $\ell_C \in \mathbb{Z}_d$ s.t.* $\Pr_{C \sim X_k}[f(\sigma(C)) = \ell_C] \geq \frac{1}{d} + \epsilon$. *There is a polynomial time algorithm (in n, m,$1/\epsilon$) that finds a mapping $\sigma' \in \mathbb{Z}_d^n$ such that σ' is $\epsilon/2$-correlated with σ with probability at least $\frac{\epsilon}{2d^2}$*

The remaining challenge in the proof of Theorem 18 is to show that there is an efficient algorithm to extract predictions of $f(\sigma(C))$ given blackbox access to an adversary $\mathcal{A}$ that breaks UF-RCA security. However, just because the adversary $\mathcal{A}$ gives the correct response to an entire password challenge $C_1, \ldots, C_\lambda$ with probability greater than $\left(\frac{1}{d} + \epsilon\right)^{\lambda}$ it does not mean that the response to each individual challenge C is correct with probability $\frac{1}{d} + \epsilon$. To obtain our predictions for individual clauses C we draw λ extra example challenge response pairs $(C'_1, f(\sigma(C'_1))), \ldots, (C'_\lambda, f(\sigma(C'_\lambda)))$, which we use to check the adversary. To obtain the label for a clause C we select a random index $i \in [\lambda]$ and give $\mathcal{A}$ the password challenge $C'_1, \ldots, C'_\lambda$, replacing C'_i with C. If for some $j < i$ the label for clause C'_j is not correct (e.g., $\neq f(\sigma(C'_j))$) then we discard the label and try again. Claim 26 in Appendix E shows that this process will give us predictions for individual clauses that are accurate with probability at least $\frac{1}{d} + \epsilon$.

6 Related Work

The literature on passwords has grown rapidly over the past decade (e.g., see [40, 46, 18, 19, 38, 15].) Perhaps most related to our paper is the work of Blocki et al. [13, 14] and Blum and Vempala [17] on developing usable and secure password management schemes. While the password management schemes proposed in these works are easier to use (e.g., involve less memorization and/or computation) than our human computable password scheme, these schemes only remain secure up to their information theoretic limit – after a very small (e.g., 1–6) number of breaches security guarantees start to break down. By contrast, our schemes remain secure after a large (e.g., 100) number or breaches.

In contrast to our work, password management software (e.g., PwdHash [49] or KeePass [48]) relies strong trust assumptions about the user's computational devices. The recent breach at LastPass[8] highlights the potential danger of such strong assumptions.

Hopper and Blum [34] designed a Human Identification Protocol based on noisy parity, a learning problem that is believed to be hard[9]. We emphasize a few fundamental differences

[8] See `https://blog.lastpass.com/2015/06/lastpass-security-notice.html/` (Retrieved 9/1/2015).
[9] Subsequent work [35, 31, 20, 36] has explored the use of the Hopper-Blum protocol for authentication on pervasive devices like smartcards.

between our work and the work of Hopper and Blum. First, a single digit challenge in their protocol consists of an n-digit vector $x \in \mathbb{Z}_{10}^n$ and the user responds with the $\mod 10$ sum of the digits at $\ell \leq n$ secret locations (occasionally the user is supposed to respond with a random digit instead of the correct response so that the adversary cannot simply use Gaussian Elimination to find the secret locations). By contrast, a single digit challenge in our protocol consists of an ordered clause of length $k \ll n$. Second, their protocols allow for an $O\left(n^{\ell/2}\right)$-time attack called Meet-In-The-Middle [34] after the adversary has seen $\tilde{O}(\log \binom{n}{\ell})$ challenge-response pairs. Thus, it is critically important to select ℓ sufficiently large (e.g., $\ell = \Omega(\log(n))$) in the Hopper-Blum protocol to defend against this Meet-In-The-Middle attack. By contrast, we focus on computation of *very simple* functions over a *constant* number of variables so that a human can compute the response to each challenge quickly. In particular, we provide strong evidence that our scheme is secure against any polynomial time attacker even if the adversary has seen up to $O\left(n^{c \cdot k}\right)$ challenge-response pairs for some constant $c \geq 1$. Finally, computations in our protocols are deterministic. This is significant because humans are not good at consciously generating random numbers [53, 27, 42] (e.g., noisy parity could be easy to learn when humans are providing source of noise)[10].

Naor and Pinkas[43] proposed using visual cryptography[44] to address a related problem: how can a human verify that a message he received from a trusted server has not been tampered with by an adversary? Their protocol requires the human to carry a visual transparency (a shared secret between the human and the trusted server in the visual cryptography scheme), which he will use to verify that messages from the trusted server have not been altered.

A related goal in cryptography, constructing pseudorandom generators in NC^0, was proposed by Goldreich [32] and by Cryan and Miltersen [23]. In Goldreich's construction we fix $C_1, \ldots, C_m \in [n]^k$ once and for all, and a binary predicate $P : \{0,1\}^k \to \{0,1\}$. The pseudorandom generator is a function $G : \{0,1\}^n \to \{0,1\}^m$, whose i'th bit $G(x)[i]$ is given by P applied to the bits of x specified by C_i. O'Donnel and Witmer gave evidence that the "Tri-Sum-And" predicate $(TSA\,(x_1, \ldots, x_5) = x_1 + x_2 + x_3 + x_4 x_5 \mod 2)$ provides near-optimal stretch. In particular, they showed that for $m = n^{1.5-\epsilon}$ Goldreich's construction with the TSA predicate is secure against subexponential-time attacks using SDP hierarchies. Our candidate human-computable password schemes use functions $f : \mathbb{Z}_{10}^k \to \mathbb{Z}_{10}$ instead of binary predicates. While our candidate functions are contained in NC^0, we note that an arbitrary function in NC^0 is not necessarily human computable.

On a technical level our statistical dimension lower bounds extend work of Feldman et al. [26], who considered the problem of finding a planted solution in a random *binary* planted constraint satisfiability problem. We extend their analysis to handle non-binary planted constraint satisfiability problems, and argue that our candidate human computable password schemes are secure. We will discuss this work in more detail later in the paper.

7 Discussion

7.1 Improving Response Time

The easiest way to improve response time is to decrease λ –the number of single-digit challenges that the user needs to solve. However, if the user wants to ensure that each of his passwords are strong enough to resist offline dictionary attacks then he would need

[10] Hopper and Blum also proposed a deterministic variant of their protocol called sum of k-mins, but this variant is *much* less secure. See additional discussion in the appendix.

to select a larger value of λ (e.g., $\lambda \geq 10$). Fortunately, there is a natural way to circumvent this problem. The user could save time by memorizing a mapping $w : \mathbb{Z}_{10} \rightarrow \{x \mid x \text{ is one of 10,000 most common english words}\}$ and responding to each challenge C with $w\left(f\left(\sigma\left(C\right)\right)\right)$ – the word corresponding to the digit $f\left(\sigma\left(C\right)\right)$. Now the user can create passwords strong enough to resist offline dictionary attacks by responding to just 3–5 challenges. Even if the adversary learns the words in the user's set he won't be able to mount online attacks. Predicting $w\left(f\left(\sigma\left(C\right)\right)\right)$ is at least as hard as predicting $f\left(\sigma\left(C\right)\right)$ even if the adversary knows the exact mapping w[11].

7.2 One-Time Challenges

Malware

Consider the following scenario: the adversary infects the user's computer with a keylogger which is never detected over the user's lifetime. We claim that it is possible to protect the user in this extreme scenario using our scheme by generating multiple (e.g., 10^6) one-time passwords for each of the user's accounts. When we initially generate the secret mapping $\sigma \sim \mathbb{Z}_d^n$ we could also generate cryptographic hashes for millions of one-time passwords $\mathbf{H}\left(\vec{C}, f_{k_1,k_2}\left(\sigma\left(\vec{C}\right)\right)\right)$. While usability concerns make this approach infeasible in a traditional password scheme (it would be far too difficult for the user to memorize a million one-time passwords for each of his accounts), it may be feasible to do this using a human computable password scheme. In our human computable password scheme we could select k_1 and k_2 large enough that $s\left(f_{k_1,k_2}\right) = \min\{k_1 + 1, (k_2 + 1)/2\} \geq 6$. Assuming that the user authenticates fewer than 10^6 times over his lifetime a polynomial time adversary would never obtain enough challenge-response examples to learn σ. The drawback is that f_{k_1,k_2} will take longer for a user to execute in his head.

Secure Cryptography in a Panoptic World

Standard cryptographic algorithms could be easily broken in a panoptic world where the user only has access to a semi-trusted computer (e.g., if a user asks a semi-trusted computer to sign a message m using a secret key sk stored on the hard drive then the computer will respond with the correct value $Sign\left(sk, m\right)$, but the adversary will learn the values m and sk). Our human computable password schemes could also be used to secure some cryptographic operations (e.g., signatures) in a panoptic world by leveraging recent breakthroughs in program obfuscation [29]. The basic idea is to obfuscate a "password locked" circuit $P_{\sigma,sk,r}$ that can sign messages under a secret key sk – we need a trusted setup phase for this step. The circuit $P_{\sigma,sk,r}$ will only sign a message m if the user provides the correct response to a unique (pseudorandomly generated) challenge for m.

7.3 Open Questions

Eliminating the Semi-Trusted Computer

Our current scheme relies on a semi-trusted computer to generate and store random public challenges. An adversary with full control over the user's computer might be able to extract the user's secret if he is able to see the user's responses to $O(n)$ adaptively selected password challenges. Can we eliminate the need for a semi-trusted computer?

[11] In fact, it is quite likely that it is much harder for the adversary to predict $w\left(f\left(\sigma\left(C\right)\right)\right)$ because the adversary will not see which word corresponds with each digit.

Exact Security Bounds

While we provided asymptotic security proofs for our human computable password schemes, it is still important to understand how much effort an adversary would need to expend to crack the secret mapping for specific values of n and m. Our attacks with a SAT solver (see Appendix C) indicate that the value $n = 26$ is too small to provide UF-RCA security even with small values of m (e.g., $m = 50$). As n increases the problem rapidly gets harder for our SAT solver (e.g., with $n = 50$ and $m = 1000$ the solver failed to find σ). We also present a public challenge with specific values of n and m to encourage cryptography and security researchers to find other techniques to attack our scheme.

References

1 Cert incident note in-98.03: Password cracking activity. http://www.cert.org/incident_notes/IN-98.03.html, July 1998. Retrieved 8/16/2011.

2 Nato site hacked. http://www.theregister.co.uk/2011/06/24/nato_hack_attack/, June 2011. Retrieved 8/16/2011.

3 Zappos customer accounts breached. http://www.usatoday.com/tech/news/story/2012-01-16/mark-smith-zappos-breach-tips/52593484/1, January 2012. Retrieved 5/22/2012.

4 Oh man, what a day! an update on our security breach. http://blogs.atlassian.com/news/2010/04/oh_man_what_a_day_an_update_on_our_security_breach.html, April 2010. Retrieved 8/18/2011.

5 Apple security blunder exposes lion login passwords in clear text. http://www.zdnet.com/blog/security/apple-security-blunder-exposes-lion-login-passwords-in-clear-text/11963, May 2012. Retrieved 5/22/2012.

6 Update on playstation network/qriocity services. http://blog.us.playstation.com/2011/04/22/update-on-playstation-network-qriocity-services/, April 2011. Retrieved 5/22/2012.

7 An update on linkedin member passwords compromised. http://blog.linkedin.com/2012/06/06/linkedin-member-passwords-compromised/, June 2012. Retrieved 9/27/2012.

8 Data breach at ieee.org: 100k plaintext passwords. http://ieeelog.com/, September 2012. Retrieved 9/27/2012.

9 Important customer security announcement. http://blogs.adobe.com/conversations/2013/10/important-customer-security-announcement.html, October 2013. Retrieved 2/10/2014.

10 Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 20–29. ACM, 1996.

11 Sam. Biddle. Anonymous leaks 90,000 military email accounts in latest antisec attack. http://gizmodo.com/5820049/anonymous-leaks-90000-military-email-accounts-in-latest-antisec-attack, July 2011. Retrieved 8/16/2011.

12 Jeremiah Blocki, Manuel Blum, and Anupam Datta. Human-computable passwords. ASIACRYPT Rump Session, 2013. URL: http://asiacrypt.2013.rump.cr.yp.to/b0279d7741ad5bab24cf5c55fd292d5c.pdf.

13 Jeremiah Blocki, Manuel Blum, and Anupam Datta. Naturally rehearsing passwords. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*,

volume 8270 of *Lecture Notes in Computer Science*, pages 361–380. Springer Berlin Heidelberg, 2013. doi:10.1007/978-3-642-42045-0_19.

14 Jeremiah Blocki, Saranga Komanduri, Lorrie Faith Cranor, and Anupam Datta. Spaced repetition and mnemonics enable recall of multiple strong passwords. In *22nd Annual Network and Distributed System Security Symposium, NDSS 2015, San Diego, California, USA, February 8-11, 2014*, 2015. URL: http://www.internetsociety.org/doc/spaced-repetition-and-mnemonics-enable-recall-multiple-strong-passwords.

15 Jeremiah Blocki, Saranga Komanduri, Ariel Procaccia, and Or Sheffet. Optimizing password composition policies. In *Proceedings of the 14th ACM Conference on Electronic Commerce*. ACM, 2013.

16 Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138. ACM, 2005.

17 Manuel Blum and Santosh Vempala. Publishable humanly usable secure password creation schemas. *Proc. of HCOMP*, 2015.

18 J. Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 538–552. IEEE, 2012.

19 S. Boztas. Entropies, guessing, and cryptography. *Department of Mathematics, Royal Melbourne Institute of Technology, Tech. Rep*, 6, 1999.

20 Julien Bringer, Hervé Chabanne, and Emmanuelle Dottax. Hbˆ+ˆ+: a lightweight authentication protocol secure against some attacks. In *Security, Privacy and Trust in Pervasive and Ubiquitous Computing, 2006. SecPerU 2006. Second International Workshop on*, pages 28–33. IEEE, 2006.

21 I.A.D. Center. Consumer password worst practices. *Imperva (White Paper)*, 2010.

22 Cheng Chu, Sang Kyun Kim, Yi-An Lin, YuanYuan Yu, Gary Bradski, Andrew Y Ng, and Kunle Olukotun. Map-reduce for machine learning on multicore. *Advances in neural information processing systems*, 19:281, 2007.

23 Mary Cryan and Peter Bro Miltersen. On pseudorandom generators in nc0. In *Mathematical Foundations of Computer Science 2001*, pages 272–284. Springer, 2001.

24 Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, July 1960. doi:10.1145/321033.321034.

25 Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

26 Vitaly Feldman, Will Perkins, and Santosh Vempala. On the complexity of random satisfiability problems with planted solutions. In *Proceedings of the 47th annual ACM symposium on Symposium on Theory of Computing*. ACM, 2015.

27 M. Figurska, M. Stanczyk, and K. Kulesza. Humans cannot consciously generate random numbers sequences: Polemic study. *Medical hypotheses*, 70(1):182–185, 2008.

28 D. Florencio and C. Herley. A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web*, pages 657–666. ACM, 2007.

29 Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 40–49. IEEE, 2013.

30 Alan E Gelfand and Adrian FM Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409, 1990.

31 Henri Gilbert, Matthew Robshaw, and Herve Sibert. Active attack against hb+: a provably secure lightweight authentication protocol. *Electronics Letters*, 41(21):1169–1170, 2005.

32 Oded Goldreich. Candidate one-way functions based on expander graphs. In *Studies in Complexity and Cryptography*. Citeseer, 2000.

33 Graham J Hitch. The role of short-term working memory in mental arithmetic. *Cognitive Psychology*, 10(3):302–323, 1978.

34 N. Hopper and M. Blum. Secure human identification protocols. *Advances in cryptology-ASIACRYPT 2001*, pages 52–66, 2001.

35 Ari Juels and Stephen A Weis. Authenticating pervasive devices with human protocols. In *Advances in Cryptology–CRYPTO 2005*, pages 293–308. Springer, 2005.

36 Jonathan Katz and Ji Sun Shin. Parallel and concurrent security of the hb and hb+ protocols. In *Advances in Cryptology-EUROCRYPT 2006*, pages 73–87. Springer, 2006.

37 Michael Kearns and Umesh Virkumar Vazirani. *An introduction to computational learning theory*. The MIT Press, 1994.

38 S. Komanduri, R. Shay, P.G. Kelley, M.L. Mazurek, L. Bauer, N. Christin, L.F. Cranor, and S. Egelman. Of passwords and people: measuring the effect of password-composition policies. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 2595–2604. ACM, 2011.

39 H. Kruger, T. Steyn, B. Medlin, and L. Drevin. An empirical assessment of factors impeding effective password management. *Journal of Information Privacy and Security*, 4(4):45–59, 2008.

40 J.L. Massey. Guessing and entropy. In *Information Theory, 1994. Proceedings., 1994 IEEE International Symposium on*, page 204. IEEE, 1994.

41 G.A. Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

42 Dave. Munger. Is 17 the "most random" number? `http://scienceblogs.com/cognitivedaily/2007/02/is_17_the_most_random_number.php`, 2007. Retrieved 8/16/2011.

43 Moni Naor and Benny Pinkas. Visual authentication and identification. In *Advances in Cryptology-CRYPTO'97*, pages 322–336. Springer, 1997.

44 Moni Naor and Adi Shamir. Visual cryptography. In *Advances in Cryptology-EUROCRYPT'94*, pages 1–12. Springer, 1995.

45 Ryan O'Donnell. Analysis of boolean functions. *Textbook in Progress. Available online at* *`http://analysisofbooleanfunctions.org/`*, 2014.

46 J. Pliam. On the incomparability of entropy and marginal guesswork in brute-force attacks. *Progress in Cryptology-INDOCRYPT 2000*, pages 113–123, 2000.

47 N. Provos and D. Mazieres. Bcrypt algorithm.

48 D Reichl. Keepass password safe, 2013. *Retrieved July*, 10, 2013.

49 Blake Ross, Collin Jackson, Nick Miyake, Dan Boneh, and John C Mitchell. Stronger password authentication using browser extensions. In *Usenix security*, pages 17–32. Baltimore, MD, USA, 2005.

50 Abe. Singer. No plaintext passwords. *;login: THE MAGAZINE OF USENIX & SAGE*, 26(7), November 2001. Retrieved 8/16/2011.

51 Saul Sternberg. Memory-scanning: Mental processes revealed by reaction-time experiments. *Cognitive psychology: Key readings*, 48, 2004.

52 Hedderik van Rijn, Leendert van Maanen, and Marnix van Woudenberg. Passing the test: Improving learning gains by balancing spacing and testing effects. In *Proceedings of the 9th International Conference of Cognitive Modeling*, 2009.

53 W.A. Wagenaar. Generation of random sequences by human subjects: A critical survey of literature. *Psychological Bulletin*, 77(1):65, 1972.

54 PA Wozniak and Edward J Gorzelanczyk. Optimization of repetition spacing in the practice of learning. *Acta neurobiologiae experimentalis*, 54:59–59, 1994.

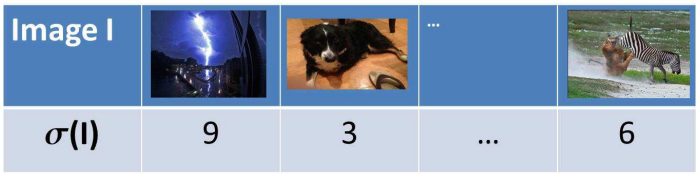


Figure 2 A Random Mapping from Images to Digits.

A Authentication Process

In this section of the appendix we illustrate our human computable password schemes graphically. In our examples, we use the function $f = f_{2,2}$. To compute the response $f(\sigma(C))$ to a challenge $C = \{x_0, \ldots, x_{13}\}$ the user computes $f(\sigma(C)) = \sigma\left(x_{\sigma(x_{10})+\sigma(x_{11}) \mod 10}\right) + \sigma(x_{12}) + \sigma(x_{13}) \mod 10$

Memorizing a Random Mapping

To begin using our human computable password schemes the user begins by memorizing a secret random mapping $\sigma : [n] \to \{0, \ldots, 9\}$ from n objects (e.g., letters, pictures) to digits. See Figure 2 for an example.

The computer can provide the user with mnemonics to help memorize the secret mapping σ – see Figures 3b and 3c. For example, if we wanted to help the user remember that $\sigma(eagle) = 2$ we would show the user Figure 3b. We observe that a $10 \times n$ table of mnemonic images would be sufficient to help the user memorize *any* random mapping σ. We stress that the computer will only save the original image (e.g., Figure 3a). The mnemonic image (e.g., Figure 3b or 3c) would be discarded after the user memorizes $\sigma(eagle)$.

Single-Digit Challenges

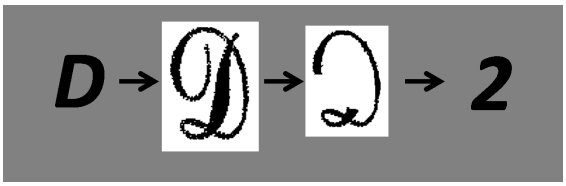
In our scheme the user computes each of his passwords by responding to a sequence of single-digit challenges. For $f = f_{2,2}$ a single-digit challenge is a tuple $C \in [n]^{14}$ of fourteen objects. See Figure 6 for an example. To compute the response $f(\sigma(C))$ to a challenge $C = \{x_0, \ldots, x_{13}\}$ the user computes $f(\sigma(C)) = \sigma\left(x_{\sigma(x_{10})+\sigma(x_{11}) \mod 10}\right) + \sigma(x_{12}) + \sigma(x_{13}) \mod 10$. Observe that this computation involves just three addition operations modulo ten. See Figure 1 for an example. In this example the response to the challenge $C = \{x_0 = \text{burger}, x_1 = \text{eagle}, \ldots, x_{10} = \text{lightning}, x_{11} = \text{dog}, x_{12} = \text{man standing on world}, x_{13} = \text{kangaroo}\}$ is

$$\begin{aligned}
f(\sigma(C)) &= \sigma\left(x_{\sigma(x_{10})+\sigma(x_{11}) \mod 10}\right) + \sigma(x_{12}) + \sigma(x_{13}) \mod 10 \\
&= \sigma\left(x_{\sigma(\text{lightning})+\sigma(\text{dog}) \mod 10}\right) \\
&\quad + \sigma(\text{man standing on world}) + \sigma(\text{kangaroo}) \mod 10 \\
&= \sigma(x_{9+3 \mod 10}) + \sigma(\text{man standing on world}) + \sigma(\text{kangaroo}) \mod 10 \\
&= \sigma(\text{minions}) + \sigma(\text{man standing on world}) + \sigma(\text{kangaroo}) \mod 10 \\
&= 7 + 4 + 5 \mod 10 = 6\,.
\end{aligned}$$

We stress that this computation is done entirely in the user's head. It takes the main author 7.5 seconds on average to compute each response.

(a) Original photo (an eagle). **(b)** Mnemonic to help the user remember $\sigma(eagle) = 2$. **(c)** Mnemonic to help the user remember $\sigma(eagle) = 6$.

Figure 3 Mnemonics to help memorize the secret mapping σ.



(a) $M_{D,2}$.

(b) $M_{D,9}$.

Figure 4 Mnemonics to help the user memorize the secret mapping σ.

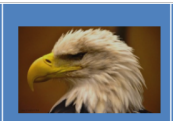
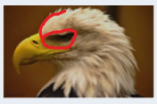
σ		...		...
...	...	...	...	...
4	The words “gold” and “beak” have four letters.	...	The words “lion” and “sand” have four letters.	...
5	The word “eagle” has five letters.	...	The words “zebra” and “grass” have five letters.	...
6		...	You can see six legs total in this picture.	...
...	...	...	...	...

Figure 5 Table of Mnemonic Helpers to Help Learn Any Secret Mapping

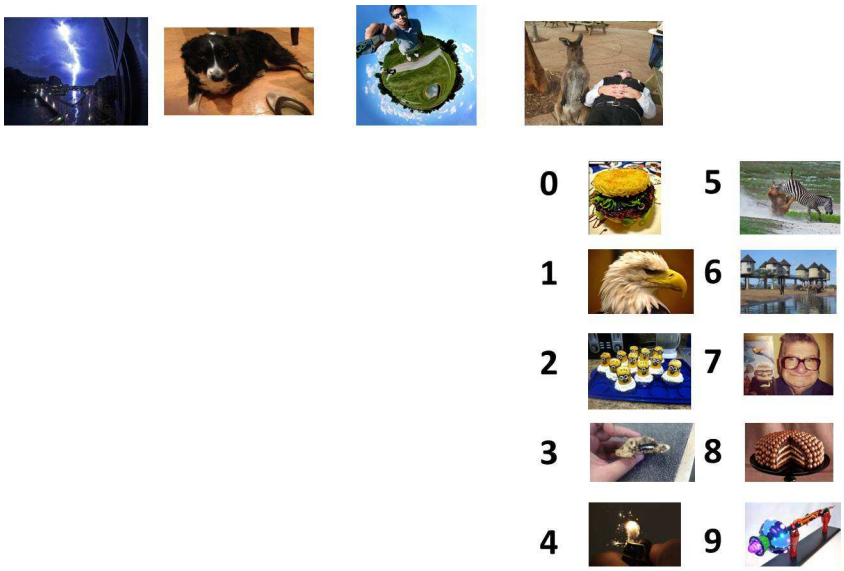


Figure 6 A single-digit challenge.

Creating an Account

To help the user create an account the computer would first pick a sequence of single-digit challenges $C_1, \ldots, C_\lambda$, where the security parameter is typically $\lambda = 10$, and would display the first challenge C_1 to the user – see Figure 7 for an example. To compute the first digit of his password the user would compute $f(\sigma(C_1))$. After the user types in the first digit $f(\sigma(C_1))$ of his password the computer will display the second challenge C_2 to the user – see Figure 8. After the user creates his account the computer will store the challenges $C_1, \ldots, C_{10}$ in public memory. The password $pw = f(\sigma(C_1)) \ldots f(\sigma(C_\lambda))$ will not be stored on the user's local computer (the authentication server may store the cryptographic hash of pw).

Authentication

Authenticating is very similar to creating an account. To help the user recompute his password for an account the computer first looks up the challenges $C_1, \ldots, C_\lambda$ which were stored in public memory, and the user authenticates by computing his password $pw = f(\sigma(C_1)) \ldots f(\sigma(C_\lambda))$. We stress that the single-digit challenges the user sees during authentication will be the same single-digit challenges that the user saw when he created the account. The authentication server verifies that the cryptographic hash of pw matches its record.

Helping the user remember his secret mapping

The computer keeps track of when the user rehearses each value of his secret mapping (e.g., $(i, \sigma(i))$ for each $i \in [n]$), and reminds the user to rehearse any part of his secret mapping that he hasn't used in a long time. One advantage of our human computable password scheme (compared with the Shared Cues scheme of Blocki et al.[13]) is that most users will use each part of their secret mapping often enough that they will not need to be reminded to rehearse – see discussion in Section A.2. The disadvantage is that we require the user to spend extra effort computing his passwords each time he authenticates.

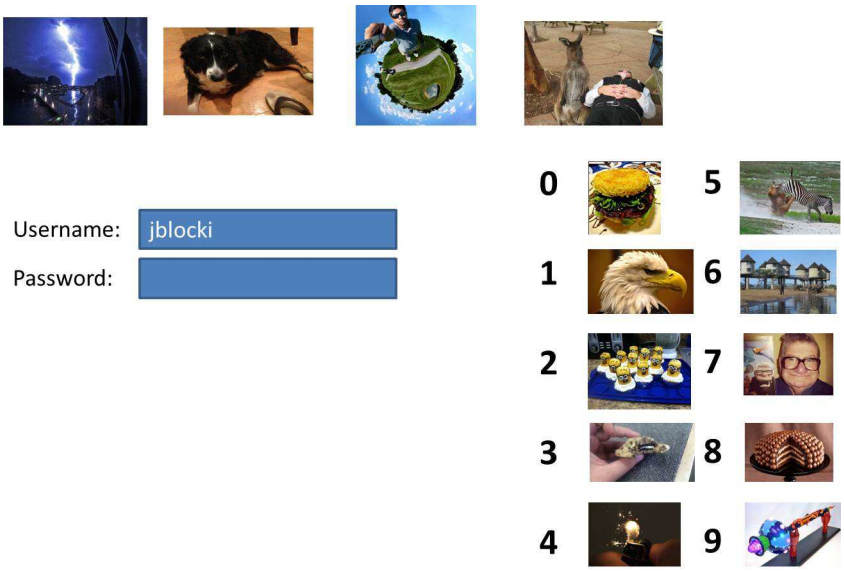


Figure 7 Login Screen

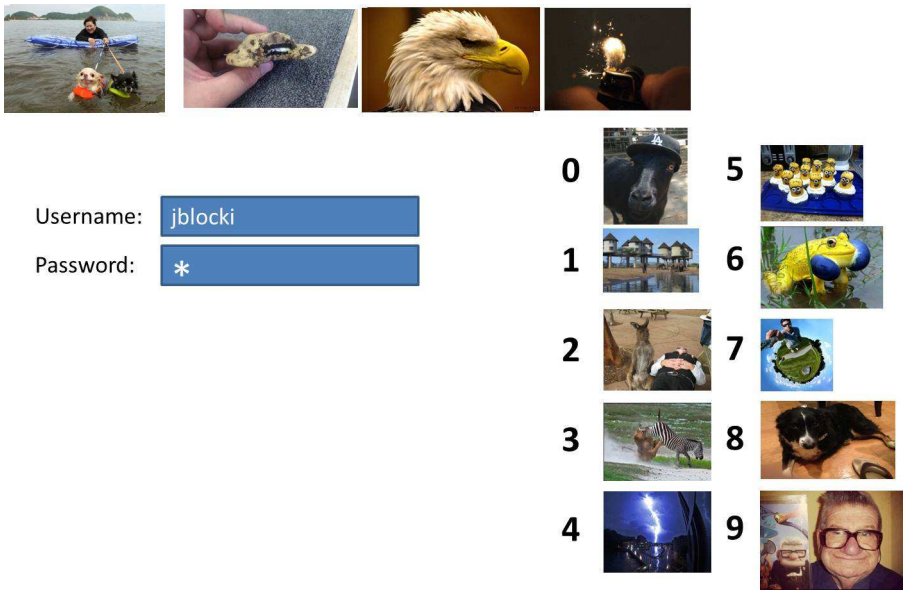


Figure 8 Login Screen after the user responds to the first single-digit challenge

Algorithm 1: MemorizeMapping

input : $I_1, ..., I_n$, d, b and $M_{i,j}$ for $i \in [n], j \in \{0, \ldots, d-1\}$.
Given $j \in \{0, \ldots, d-1\}$ and $i \in \{1, \ldots, n\}$ $M_{i,j}$ is a mnemonic to help the user associate image I_i with the number j. d is whatever base the user is familiar with (typically $d = 10$), and the value b contains random bits which are used to select the secret mapping σ. ;

Generate and Memorize Secret Mapping;
for $i \leftarrow 1$ **to** n **do**
  `// Using random bits b`
  $\sigma(i) \sim \{0, \ldots, d-1\}$;
  $M_i \leftarrow M_{i,\sigma(i)}$;
  (User) Using M_i memorizes the association $(I_i, \sigma(i))$ for $i \in [n]$. ;
end

A.1 Formal View

We now present a formal overview of the authentication process. Algorithm 1 outlines the initialization process in which the user memorizes a secret random mapping σ generated by the user's computer, and Algorithm 2 outlines the account creation process. In Algorithm 2 the user generates the password for an account i by computing the response to a sequence of random challenges C generated by the user's computer. The sequence of challenges are stored in public memory. We assume that all steps in algorithms 1, 2 and 3 are executed on the user's local computer unless otherwise indicated. We also assume that the initialization phase (Algorithms 1 and 2) is carried out in secret (e.g., we assume that the secret mapping is chosen in secret), but we do not assume that the challenges are kept secret. We use **(User)** to denote a step completed by the human user and we use **(Server)** to denote a step completed by a third-party server. Algorithm 3 illustrates the authentication process.

A.2 Memorizing and Rehearsing σ

After the user memorizes σ he may need to rehearse parts of the mapping periodically to ensure that he does not forget it. How much effort does this require? Blocki et al.[13] introduced a usability model to estimate how much extra effort that a user would need to spend rehearsing the mapping σ. We used this model to obtain the predictions in Table 1.

Imagine that we had a program keep track of how often the user rehearsed each association $(i, \sigma(i))$ and predict how much longer the user will safely remember the association $(i, \sigma(i))$ without rehearsing again – updating this prediction after each rehearsal. The user rehearses the association $(i, \sigma(i))$ *naturally* whenever he needs to recall the value of $\sigma(i)$ while computing the password for any of his password protected accounts. If the user is in danger of forgetting the value $\sigma(i)$ then the program sends the user a reminder to rehearse (Blocki et al.[13] call this an *extra rehearsal*). Table 1 shows the value of $\mathbb{E}[ER_{365}]$, the expected number of extra rehearsals that the user will be required to do to remember the secret mapping σ during the first year. This value will depend on how often the user rehearses σ naturally. We consider four types of users: Active, Typical, Occasional and Infrequent. An Active user visits his accounts more frequently than a Infrequent user. See Appendix H for specific details about how Table 1 was computed.

Algorithm 2: CreateChallenge

input : n, i, λ, d, b and $I_1, ..., I_n$.

Generate Password Challenge for account A_i. t is a security parameter which specifies how many digits the password will contain.;

for $j = 1 \to \lambda$ **do**

  $C_j^i \sim X_k$;

  `// Using random bits b`

end

$\vec{C}_i \leftarrow \langle C_1^i, \ldots, C_\lambda^i \rangle$;

Store record $\left(i, \vec{C}_i\right)$;

for $j = 1 \to \lambda$ **do**

  Load images from C_j^i. ;

  Display C_j^i for the user. ;

  (User) Computes $q_j \leftarrow f\left(\sigma\left(C_i^j\right)\right)$;

end

Send $\langle q_1, \ldots, q_\lambda \rangle = f\left(\sigma\left(\vec{C}_i\right)\right)$ to server i;

`// H is a strong cryptographic hash function`

(Server i) Stores $h_i = \mathbf{H}\left(\vec{C}_i, \langle q_1, \ldots, q_\lambda \rangle\right)$;

Algorithm 3: Authenticate

input : Security parameter λ. Account $i \in [m]$. Challenges $\vec{C}_1, \ldots, \vec{C}_m$.

$\langle C_1^i, \ldots, C_\lambda^i \rangle \leftarrow \vec{C}_i$;

`// Display Single Digit Challenges`

for $j = 1 \to \lambda$ **do**

  (Semi-Trusted Computer) Load images from C_j^i.;

  (Semi-Trusted Computer) Displays C_j^i to the user. ;

  (User) Computes $q_j \leftarrow f\left(\sigma\left(C_i^j\right)\right)$. ;

end

(Semi-Trusted Computer) Sends $\langle q_1, \ldots, q_\lambda \rangle$ to the server for account i. ;

(Server) Verifies that $\mathbf{H}\left(\vec{C}_i, \langle q_1, \ldots, q_\lambda \rangle\right) = h_i$;

Table 1 $\mathbb{E}[ER_{365}]$: Extra Rehearsals over the first year to remember σ in our scheme with $f_{2,2}$ or $f_{1,3}$. Compared with Shared Cues schemes SC-0,SC-1 and SC-2[13].

	Our Scheme ($\sigma \in \mathbb{Z}_{10}^n$)			Shared Cues		
User	$n = 100$	$n = 50$	$n = 30$	SC-0	SC-1	SC-2
Very Active	0.396	0.001	≈ 0	≈ 0	3.93	7.54
Typical	2.14	0.039	≈ 0	≈ 0	10.89	19.89
Occasional	2.50	0.053	≈ 0	≈ 0	22.07	34.23
Infrequent	70.7	22.3	6.1	≈ 2.44	119.77	173.92

Table 2 Single-Digit Challenge Layout in Scheme 1.

A	B	C	D
0	E	5	J
1	F	6	K
2	G	7	L
3	H	8	M
4	I	9	N

Table 3 Human Computable Password Challenges
n – Secret Length
m – # Challenge-Response Pairs

	Scheme 1 ($f_{2,2}$)		Scheme 2 ($f_{1,3}$)	
n	m	Winner	m	Winner
100 digits	1000	N/A	500	N/A
	500	N/A	300	N/A
	300	N/A	200	N/A
50 digits	500	N/A	300	N/A
	300	N/A	150	N/A
	150	N/A	100	N/A
30 digits	300	CSP Solver	150	N/A
	100	N/A	100	N/A
	50	N/A	50	N/A

Discussion

One of the advantages of our human computable passwords schemes is that memorization is essentially a one time cost for our Very Active, Typical and Occasional users. Once the user has memorized the mapping $\sigma : \{1, ..., n\} \to \mathbb{Z}_d$ he will get sufficient natural rehearsal to maintain this memory. In fact, our schemes require the user to expend *less* extra effort rehearsing his secret mapping than the Shared Cues password management scheme of Blocki et al. [13] (with the exception of SC-0 – the least secure Shared Cues scheme). Intuitively, this is because human computable password schemes require to recall $\sigma(i)$ for multiple different values of i to respond to each single-digit challenge C. To compute $f_{2,2}(\sigma(\{0, \ldots, 13\}))$ the user would need to recall the values of $\sigma(10), \sigma(11), \sigma(12), \sigma(13)$ and $\sigma(j)$, where $j = \sigma(10) + \sigma(11) \mod 10$. If the user has 10 digit passwords then he will naturally rehearse the value of $\sigma(i)$ for up to fifty different values of i each time he computes one of his passwords. While the user needs to spend extra time computing his password each time he authenticates in our human computable password scheme, this extra computation time gives the user more opportunities to rehearse his secret mapping.

B Human Computable Passwords Challenge

While we provided asymptotic security bounds for our human computable password schemes in our context it is particularly important to understand the constant factors. In our context, it may be reasonable to assume that $n \leq 100$ (e.g., the user may be unwilling to memorize longer mappings). In this case it would be feasible for the adversary to execute an attack that takes time proportional to $10^{\sqrt{n}} \leq 10^{10}$. We conjecture that in practice scheme 2 ($f_{1,3}$) is slightly weaker than scheme 1 ($f_{2,2}$) when $n \leq 100$ despite the fact that $s(f_{2,2}) < s(f_{1,3})$

because of the attack described in Remark G.2. This attack requires $\tilde{O}\left(n^{1+g(f)/2}\right)$ examples, and the running time $O\left(10^{\sqrt{n}}n^3\right)$ may be feasible for $n \leq 100$. To better understand the exact security bounds we created several public challenges for researchers to break our human computable password schemes under different parameters (see Table 3). At this time these challenges remain unsolved even after they were presented during the rump sessions at a cryptography conference and a security conference[12]. The challenges can be found at `http://www.cs.cmu.edu/~jblocki/HumanComputablePasswordsChallenge/challenge.htm`. For each challenge we selected a random secret mapping $\sigma \in \mathbb{Z}_{10}^n$, and published (1) m single digit challenge-response pairs $(C_1, f(\sigma(C_1))), \ldots, (C_m, f(\sigma(C_m)))$, where each clause C_i is chosen uniformly at random from X_k, and (2) 20 length – 10 password challenges $\vec{C}_1, \ldots, \vec{C}_{20} \in (X_k)^{10}$. The goal is to guess one of the secret passwords $p_i = f\left(\sigma\left(\vec{C}_i\right)\right)$ for some $i \in [20]$.

C CSP Solver Attacks

Theorems 14 and 18 provide asymptotic security bounds (e.g., an adversary needs to see $m = \tilde{\Omega}\left(n^{s(f)}\right)$ challenge-response pairs to forge passwords). However, in our context n is somewhat small (e.g., $n \leq 100$). Thus, it is also important to address the following question: how many challenge-response pairs does the adversary need to see before it becomes feasible for the adversary to recover the secret on a modern computer? To better understand the exact security bounds of our human computable password schemes we used a Constraint Satisfaction Problem (CSP) solver to attack our scheme. We also created several public challenges to break our candidate human computable password schemes (see Table 3).

CSP Solver

Our computations were performed on a computer with a 2.83 GHz Intel Core2 Quad CPU and 4 GB of RAM. In each instance we generated a random mapping $\sigma : [n] \to \mathbb{Z}_{10}$ and m random challenge response pairs $(C, f(\sigma(C)))$ using the functions $f_{2,2}$ and $f_{1,3}$. We used the Constraint Satisfaction Problem solver from the Microsoft Solver Foundations library to try to solve for σ[12]. The results of this attack are shown in Tables 4 and 5. Due to limited computational resources we terminated each instance if the solver failed to find the secret mapping within 2.5 days, and if our solver failed to find σ in 2.5 days on and instance (n, m) we did not run the solver on strictly harder instances (e.g., (n', m') with $n' \geq n$ and $m' \leq m$). We remark that our empirical results are consistent with the hypothesis that the time/space resources consumed by the CSP solver increase exponentially in n (e.g., when we decrease n from 30 to 26 with $m = 100$ examples the CSP solver can solve for $\sigma \in \mathbb{Z}_{10}^{26}$ in 40 minutes, while the solver failed to find $\sigma \in \mathbb{Z}_{10}^{30}$ in 2.5 days and we observe similar threshold behavior in other columns of the table.).

D Statistical Dimension

At a high level our statistical dimension lower bounds closely mirror the lower bounds from [26] for binary predicates. For example, Lemmas 21, 22, 15, 23 and 25 are similar to Lemmas 2, 4, 5, 6 and 7 from [26] respectively.

[12] http://blogs.msdn.com/b/solverfoundation/ (Retrieved 9/15/2014).

Table 4 CSP Solver Attack on $f_{2,2}$
Key: UNSOLVED – Solver failed to find solution in 2.5 days; HARD – Instance is harder than an unsolved instance;

	$m = 50$	$m = 100$	$m = 300$	$m = 500$	$m = 1000$	$m = 10000$
$n = 26$	23.5 hr	40 min	4.5 hr	29 min	10 min	2 min
$n = 30$	HARD	UNSOLVED	2.33 hr	35.5 min	10 min	20 s
$n = 50$	HARD	HARD	HARD	HARD	UNSOLVED	7 hr
$n = 100$	HARD	HARD	HARD	HARD	HARD	UNSOLVED

Table 5 CSP Solver Attack on $f_{1,3}$
Key: UNSOLVED – Solver failed to find solution in 2.5 days; HARD – Instance is harder than an unsolved instance;

	$m = 50$	$m = 100$	$m = 300$	$m = 500$	$m = 1000$	$m = 10000$
$n = 26$	8.7 hr	53 min	1.33 hr	13.5 min	6.3min	2 min
$n = 30$	HARD	UNSOLVED	1 hr	41 min	2 min	15 s
$n = 50$	HARD	HARD	HARD	HARD	UNSOLVED	6.5 hr
$n = 100$	HARD	HARD	HARD	HARD	HARD	UNSOLVED

While the high level proof strategy is very similar, we stress that our lower bounds do requires new ideas because we are working with planted solutions $\sigma \in \mathbb{Z}_d^n$ instead of $\sigma \in \mathbb{Z}_d^n$. We use the basis functions χ_α where for $\alpha \in \mathbb{Z}_d^n$ is

$$\chi_\alpha(x) = \exp\left(\frac{-2\pi\sqrt{-1}\,(x \cdot \alpha)}{d}\right) .$$

Note that if $d > 2$ then the Fourier coefficients $\hat{b}_\alpha$ of a function $b : \mathbb{Z}_d^k \to \mathbb{R}$ might include complex numbers. While we need to take care to deal with the possibility that a Fourier coefficients might be complex, we are still able to apply powerful tools from Fourier analysis. For example, Parseval's identity

$$\sum_{\alpha \in \mathbb{Z}_d^k} \left|\hat{b}_\alpha\right|^2 = \mathbb{E}_{x \sim \mathbb{Z}_d^k}\left[b(x)^2\right] ,$$

still applies and there are versions of the hypercontractivity theorem [45, Chapter 10] that still apply even when Fourier coefficients are complex.

Another difference is that the reference distribution is defined over clauses and outputs $X_k \times \mathbb{Z}_d$ (instead of just clauses X_k) because we are working with a function $f : \mathbb{Z}_d^n \to \mathbb{Z}_d$ with non-binary outputs. Some care is needed in finding the right reference distribution. Unlike [26] we cannot just use the uniform distribution over $X_k \times \mathbb{Z}_d$ as a reference distribution – instead the reference distribution inherently depends on the function f (Of course it is must still be independent of σ).

The following definition will be useful in our proofs.

▶ **Definition 20.** Given a clause $C \in X_k$ and $S \subseteq [k]$ of size ℓ, we let $C_{|S} \in X_\ell$ denote the clause of variables of C at the positions with indices in S (e.g., if $C = (1, \ldots, k)$ and $S = \{1, 5, k-2\}$ then $C_{|S} = (1, 5, k-2) \in X_3$). Given a function $h : X_k \times \mathbb{Z}_d \to \mathbb{R}$, a clause $C_\ell \in X_\ell$ and $i \in \mathbb{Z}_d$ and a set $S \subseteq [k]$ of size $|S| = \ell$ we define

$$h_S^i(C_\ell) = \frac{|X_\ell|}{|X_k|} \sum_{C \in X_k, C_{|S} = C_\ell} h(C, i) .$$

We first show that $\Delta(\sigma, h)$ can be expressed in terms of the Fourier coefficients of $\hat{Q}$ as well as the functions h_ℓ. In particular, given a function $h : X_k \times \mathbb{Z}_d \to \mathbb{R}$ we define the degree ℓ function $b_\ell : \mathbb{Z}_d^n \to \mathbb{C}$ as follows

$$b_\ell(\sigma) \doteq \frac{1}{2} \sum_{\alpha \in \mathbb{Z}_d^\ell : H(\alpha) = \ell} \binom{k}{\ell} \sum_{i=0}^{d-1} \hat{Q}_\alpha^{f,i} \sum_{C \in X_\ell} \chi_\alpha(\sigma(C)) h_\ell^i(C) \ .$$

Lemma 21 states that

$$\Delta(\sigma, h) = \sum_{\ell=1}^{k} \frac{1}{|X_\ell|} b_\ell(\sigma) \ .$$

This observation will be important later because we can use hypercontractivity to bound the expected value of $|b_\ell(\sigma)|$. Notice that if Q has distributional complexity r and $\ell \leq r$ then $b_\ell(\sigma) = 0$ because $\hat{Q}_\alpha^{f,i} = 0$ for all $i \in \mathbb{Z}_d$ and $\alpha \in \mathbb{Z}_d^k$ s.t. $1 \leq H(\alpha) \leq r$. This means that first r terms of the sum in Lemma 21 will be zero.

▶ **Lemma 21.** *For every $\sigma \in \mathbb{Z}_d^k$, $j \in \mathbb{Z}_d$ and $h : X_k \to \mathbb{R}$ we have $\Delta(\sigma, h) = \sum_{\ell=1}^{k} \frac{1}{|X_\ell|} b_\ell(\sigma)$.*

Proof of Lemma 21. Before calculating we first observe that for any $j \in \mathbb{Z}_d$ we have

$$\hat{Q}_{\vec{0}}^{f,j} = \mathbb{E}_{x \sim \mathbb{Z}_d^k}\left[Q^{f,j}(x) \chi_\alpha(x)\right] = \mathbb{E}_{x \sim \mathbb{Z}_d^k}\left[Q^{f,j}(x)\right] = \sum_{x \in \mathbb{Z}_d^k} \frac{Q^{f,j}(x)}{d^k} = \frac{2-d}{d} \ .$$

Given $\alpha \in \mathbb{Z}_d^k$ we also define $S_\alpha \subset [k]$ to be the set of indices i s.t. $\alpha_i \neq 0$ – $|S_\alpha| = H(\alpha)$.

Now we note that

$$\begin{aligned}
\Delta(\sigma,h) &= \mathbb{E}_{C\sim X_k}\left[h\left(C,f\left(\sigma\left(C\right)\right)\right)\right]-\mathbb{E}_{(C,i)\sim T}\left[h\left(C,i\right)\right]\\
&= \sum_{C\in X_k}\left(\frac{h\big(C,f(\sigma(C))\big)}{|X_k|}-\sum_{i=0}^{d-1}\Pr_T[(C,i)]h(C,i)\right)\\
&= \sum_{C\in X_k}\left(\frac{h\big(C,f(\sigma(C))\big)}{|X_k|}-\sum_{i=0}^{d-1}\frac{\Pr_{x\sim\mathbb{Z}_d^k}[f(x)=i]}{|X_k|}h(C,i)\right)\\
&= \sum_{C\in X_k}\sum_{i=0}^{d-1}\left(\frac{h(C,i)\left(\frac{Q_\sigma^{f,i}(C)+1}{2}-\Pr_{x\sim\mathbb{Z}_d^k}[f(x)=i]\right)}{|X_k|}\right)\\
&= \sum_{C\in X_k}\sum_{i=0}^{d-1}\left(\frac{h(C,i)}{|X_k|}\left(\frac{1}{2}-\Pr_{x\sim\mathbb{Z}_d^k}[f(x)=i]+\sum_{\alpha\in\mathbb{Z}_d^k}\frac{\hat{Q}_\alpha^{f,i}\chi_\alpha\left(\sigma\left(C\right)\right)}{2}\right)\right)\\
&= \sum_{C\in X_k}\sum_{i=0}^{d-1}\left(\frac{h(C,i)}{|X_k|}\left(\frac{1}{d}-\Pr_{x\sim\mathbb{Z}_d^k}[f(x)=i]+\sum_{\alpha\in\mathbb{Z}_d^k:\alpha\neq\vec{0}}\frac{\hat{Q}_\alpha^{f,i}\chi_\alpha\left(\sigma\left(C\right)\right)}{2}\right)\right)\\
&= \sum_{C\in X_k}\sum_{i=0}^{d-1}\left(\frac{h(C,i)}{|X_k|}\left(\sum_{\alpha\in\mathbb{Z}_d^k:\alpha\neq\vec{0}}\frac{\hat{Q}_\alpha^{f,i}\chi_\alpha\left(\sigma\left(C\right)\right)}{2}\right)\right)\\
&= \sum_{C\in X_k}\sum_{i=0}^{d-1}\left(\frac{h(C,i)}{|X_k|}\left(\sum_{\ell=1}^{k}\sum_{\alpha\in\mathbb{Z}_d^k:H(\alpha)=\ell}\frac{\hat{Q}_\alpha^{f,i}\chi_\alpha\left(\sigma\left(C\right)\right)}{2}\right)\right)\\
&= \frac{1}{2\left|X_k\right|}\sum_{\ell=1}^{k}\sum_{C\in X_k}\sum_{i=0}^{d-1}\left(h(C,i)\left(\sum_{\alpha\in\mathbb{Z}_d^k:H(\alpha)=\ell}\hat{Q}_\alpha^{f,i}\chi_\alpha\left(\sigma\left(C\right)\right)\right)\right)\\
&= \frac{1}{2\left|X_k\right|}\sum_{\ell=1}^{k}\sum_{\alpha\in\mathbb{Z}_d^k:H(\alpha)=\ell}\sum_{C\in X_k}\sum_{i=0}^{d-1}\left(h(C,i)\left(\hat{Q}_\alpha^{f,i}\chi_\alpha\left(\sigma\left(C\right)\right)\right)\right)\\
&= \frac{1}{2\left|X_k\right|}\sum_{\ell=1}^{k}\sum_{\alpha\in\mathbb{Z}_d^k:H(\alpha)=\ell}\sum_{i=0}^{d-1}\hat{Q}_\alpha^{f,i}\sum_{C_\ell\in X_\ell}\sum_{C\in X_k,C_{|S_\alpha}=C_\ell}\chi_\alpha\left(\sigma\left(C\right)\right)h^i(C)\\
&= \frac{1}{2\left|X_k\right|}\sum_{\ell=1}^{k}\sum_{\alpha\in\mathbb{Z}_d^\ell:H(\alpha)=\ell}\binom{k}{\ell}\sum_{i=0}^{d-1}\hat{Q}_\alpha^{f,i}\sum_{C_\ell\in X_\ell}\sum_{C\in X_k,C_{|S_\alpha}=C_\ell}\chi_\alpha\left(\sigma\left(C_\ell\right)\right)h(C,i)\\
&= \sum_{\ell=1}^{k}\frac{1}{2\left|X_\ell\right|}\sum_{\alpha\in\mathbb{Z}_d^\ell:H(\alpha)=\ell}\binom{k}{\ell}\sum_{i=0}^{d-1}\hat{Q}_\alpha^{f,i}\sum_{C_\ell\in X_\ell}\chi_\alpha\left(\sigma\left(C_\ell\right)\right)h_{S_\alpha}^i(C)\\
&= \sum_{\ell=1}^{k}\frac{1}{|X_\ell|}b_\ell(\sigma)\ .
\end{aligned}$$

◀

The following lemma is similar to Lemma 4 from [26]. Lemma 22 is based on the general hypercontractivity theorem [45, Chapter 10] and applies to more general (non-boolean) functions.

▶ **Lemma 22.** *[45, Theorem 10.23] If $b : \mathbb{Z}_d^n \to \mathbb{R}$ has degree at most ℓ then for any $t \geq \left(\sqrt{2ed}\right)^\ell$,*

$$\Pr_{x\sim\mathbb{Z}_d^n}\left[|b(x)| \geq t\|b\|_2\right] \leq \frac{1}{d^\ell}\exp\left(-\frac{\ell}{2ed}t^{2/\ell}\right) ,$$

where $\|b\|_2 = \sqrt{\mathbb{E}_{x\sim\mathbb{Z}_d^n}\left[b\left(x\right)^2\right]}$

Lemma 15 and its proof are almost identical to Lemma 5 in [26]. We simply replace their concentration bounds with the concentration bounds in Lemma 22. We include the proof for completeness.

▶ **Lemma 15** (restated). *Let $b : \mathbb{Z}_d^n \to \mathbb{R}$ be any function with degree at most ℓ, and let $\mathcal{D}' \subseteq \mathbb{Z}_d^n$ be a set of assignments for which $d' = d^n/|\mathcal{D}'| \geq e^\ell$. Then $\mathbb{E}_{\sigma\sim\mathcal{D}'}\left[|b\left(\sigma\right)|\right] \leq 2(\ln d'/c_0)^{\ell/2}\|b\|_2$, where $c_0 = \ell\left(\frac{1}{2ed}\right)$ and $\|b\|_2 = \sqrt{\mathbb{E}_{x\sim\mathbb{Z}_d^n}\left[b\left(x\right)^2\right]}$.*

Proof of Lemma 15. The set $\mathcal{D}'$ contains $1/d'$ fraction of points in $\mathbb{Z}_d^n$. Therefore,

$$\Pr_{x\sim\mathcal{D}'}\left[|b(x)| \geq t\|b\|_2\right] \leq \frac{d'}{d^\ell}\exp\left(-\frac{\ell}{2ed}t^{2/\ell}\right) ,$$

for any $t \geq \left(\sqrt{2ed}\right)^\ell$. For any random variable Y and value $a \in \mathbb{R}$,

$$\mathbb{E}[Y] \leq a + \int_a^\infty \Pr[Y \geq t]\, dt .$$

We set $Y = |b\left(\sigma\right)|/\|b\|_2$ and $a = \left(\frac{\ln d'}{c_0}\right)^{\ell/2}$. Assuming that $a > \left(\sqrt{2ed}\right)^\ell$ we get

$$\begin{aligned}\frac{\mathbb{E}_{\sigma\sim\mathcal{D}'}[|b(\sigma)|]}{\|b\|_2} &\leq \left(\frac{\ln d'}{c_0}\right)^{\ell/2} + \int_{(\ln d'/c_0)^{\ell/2}}^\infty \frac{d'}{d^\ell}\cdot e^{-c_0 t^{2/\ell}} dt \\ &= (\ln d'/c_0)^{\ell/2} + \frac{\ell\cdot d'}{2d^\ell\cdot c_0^{\ell/2}}\cdot\int_{\ln d'}^\infty e^{-z} z^{\ell/2-1} dz .\end{aligned}$$

Let $u(i) \doteq \int_{\ln d'}^\infty e^{-z} z^{\ell/2-i} dz$. Applying integration by parts we have

$$\begin{aligned}u(i) \doteq \int_{\ln d'}^\infty e^{-z} z^{\ell/2-i} dz &= \left(e^{-z} z^{\ell/2-i+1}\right)\Big|_{\ln d'}^\infty + \int_{\ln d'}^\infty e^{-z} z^{\ell/2-i+1} dz - \left(\frac{\ell}{2}-i\right)\int_{\ln d'}^\infty e^{-z} z^{\ell/2-i} dz \\ &= \left(e^{-z} z^{\ell/2-i+1}\right)\Big|_{\ln d'}^\infty + u(i-1) - \left(\frac{\ell}{2}-i\right)u(i) .\end{aligned}$$

Thus,

$$u(i-1) = \left(\frac{\ell}{2}-i+1\right)u(i) - \left(e^{-z} z^{\ell/2-i+1}\right)\Big|_{\ln d'}^\infty = \left(\frac{\ell}{2}-i+1\right)u(i) + \frac{(\ln d')^{\frac{\ell}{2}-i+1}}{d'} .$$

Unrolling the recurrence for $T \geq 1$ we get

$$u(1) = \frac{(\ln d')^{\frac{\ell}{2}}}{d'} + \sum_{i=1}^{T-1}\frac{(\ln d')^{\frac{\ell}{2}-i}}{d'}\prod_{j=1}^{i}\left(\frac{\ell}{2}-j+1\right) + u\left(T+1\right)\prod_{j=1}^{T}\left(\frac{\ell}{2}-j+1\right) .$$

We also note that for $T = \lceil \frac{\ell}{2} + 1 \rceil$ we have $u(T+1)\prod_{j=1}^{T}\left(\frac{\ell}{2} - j + 1\right) \leq 0$. This follows because $u(T+1) \geq 0$ for any integer $T \geq 0$ and $\prod_{j=1}^{T}\left(\frac{\ell}{2} - j + 1\right) \leq 0$. It follows that

$$u(1) \leq \frac{(\ln d')^{\frac{\ell}{2}}}{d'} + \sum_{i=1}^{\lceil \frac{\ell}{2}+1 \rceil} \frac{(\ln d')^{\frac{\ell}{2}-i}}{d'} \prod_{j=1}^{i}\left(\frac{\ell}{2} - j + 1\right) \leq \frac{(\ln d')^{\frac{\ell}{2}}}{d'}\left(1 + \sum_{i=1}^{\lceil \frac{\ell}{2}+1 \rceil} \ell^{-i}\left(\frac{\ell}{2}\right)^{i}\right) \leq \frac{2\,(\ln d')^{\frac{\ell}{2}}}{d'}$$

where we used the condition $d' \geq e^{\ell}$ to obtain the second to last inequality. Now we have

$$\begin{aligned}
\frac{\mathbb{E}_{\sigma\sim\mathcal{D}'}[|b(\sigma)|]}{\|b\|_2} &\leq (\ln d'/c_0)^{\ell/2} + \frac{\ell \cdot d'}{2d^{\ell} \cdot c_0^{\ell/2}} \cdot u(1) \\
&\leq (\ln d'/c_0)^{\ell/2} + \frac{\ell \cdot d'}{2d^{\ell} \cdot c_0^{\ell/2}} \cdot \left(\frac{2\,(\ln d')^{\frac{\ell}{2}}}{d'}\right) \\
&\leq 2(\ln d'/c_0)^{\ell/2} \,.
\end{aligned}$$

◀

▶ **Lemma 23.** *Let $\mathcal{D}' \subseteq \{0,\ldots,d-1\}^n$ be a set of assignments for which $d' = d^n/|\mathcal{D}'|$. Then*

$$\mathbb{E}_{\sigma\sim\mathcal{D}'}\left[\left|\frac{1}{|X_\ell|} b_\ell(\sigma)\right|\right] \leq 2\left(\binom{k}{\ell}\sqrt{\ell! d}\right)(\ln d'/c_0)^{\ell/2} \max_{\alpha\in\mathbb{Z}_d^\ell} \sum_{i=0}^{d-1} \frac{\|h_{S_\alpha}(\sigma)\|_2}{\sqrt{|X_\ell|}} \,.$$

Proof. For simplicity of notation we set $b = b_\ell$. Our first goal will be to find the Fourier coefficients of

$$b(\sigma) = \frac{1}{2} \sum_{\alpha\in\mathbb{Z}_d^\ell : H(\alpha)=\ell} \binom{k}{\ell} \sum_{i=0}^{d-1} \hat{Q}_\alpha^{f,i} \sum_{C\in X_\ell} \chi_\alpha(\sigma(C))\, h_{S_\alpha}^i(C) \,.$$

Given $\alpha = (\alpha_1,\ldots,\alpha_\ell) \in \mathbb{Z}_d^\ell$ with $H(\alpha) = \ell$ and a clause $C = (c_1,\ldots,c_\ell) \in X_\ell$ we define the projection $\alpha^C \in \mathbb{Z}_d^\ell$ of α onto C to be the unique vector s.t. $\alpha^C_{c_i} = \alpha_i$ for each $i \leq \ell$ and $\alpha_j = 0$ for each $j \notin \{c_1,\ldots,c_\ell\}$ – note that $H(\alpha^C) = \ell$.

Given $\alpha, \alpha' \in \mathbb{Z}_d^k$ with $H(\alpha) = H(\alpha') = \ell$ and $C, C \in X_\ell$ we say that the pairs (α, C_ℓ) and (α', C'_ℓ) are equivalent if and only if their projections are equal $\alpha^C = \alpha'^{C'}$[13]. We can partition the set $\{\alpha \in \mathbb{Z}_d^k : H(\alpha) = \ell\} \times X_\ell$ into equivalence classes $E_1,\ldots,E_t$. If the pairs (α, C_ℓ) and (α', C'_ℓ) are equivalent then we observe that the clauses C and C' must contain the same variables though perhaps in a different order. Furthermore, given an equivalence class E_j such that $(\alpha, C) \in E_j$ we have $(\alpha', C) \notin E_j$ for any $\alpha' \neq \alpha \in \mathbb{Z}_d^k$. Thus each equivalence class has size $\ell!$ because there are $\ell!$ ways to reorder the ℓ variables in a clause C. We can rewrite $b(\sigma)$ as

$$b(\sigma) = \frac{\binom{k}{\ell}}{2} \sum_{j=1}^{t} \sum_{(\alpha,C)\in E_j} \sum_{i=0}^{d-1} \hat{Q}_\alpha^{f,i} h_{S_\alpha}^i(C) \chi_\alpha(\sigma(C))$$

Let $(\alpha, C) \in E_j$ then the Fourier coefficient of α^C is

$$\hat{b}_j = \hat{b}_{\alpha^C} = \frac{\binom{k}{\ell}}{2} \sum_{(\alpha',C')\in E_j} \sum_{i=0}^{d-1} \hat{Q}_\alpha^{f,i} h_\ell^i(C') \,.$$

[13] For example, if $C = (1,2,5)$, $C = (1,5,2)$ and $\alpha = (4,5,6)$ and $\alpha' = (4,6,5)$ then the pairs (α, C_ℓ) and $\left(\alpha', C'_\ell\right)$ are equivalent.

We also note that $t = \frac{|X_\ell|(d-1)^k}{\ell!}$.

Now we can apply Parseval's identity along with the Cauchy-Schwarz inequality to obtain

$$\begin{aligned}
\mathbb{E}_{\sigma\sim\mathbb{Z}_d^n}\left[b(\sigma)\overline{b(\sigma)}\right] &= \mathbb{E}_{\sigma\sim\mathbb{Z}_d^n}\left[|b(\sigma)|^2\right] \\
&= \mathbb{E}_{\sigma\sim\mathbb{Z}_d^n}\left[\sum_{j=1}^{t}\left|\hat{b}_j\right|^2\right] \\
&= \frac{\binom{k}{\ell}^2}{4}\mathbb{E}_{\sigma\sim\mathbb{Z}_d^n}\left[\sum_{j=1}^{t}\left|\sum_{(C,\alpha)\in E_j}\sum_{i=0}^{d-1}\hat{Q}_\alpha^{f,i}h_{S_\alpha}^i(C)\right|^2\right] \\
&\le \frac{\binom{k}{\ell}^2}{4}\mathbb{E}_{\sigma\sim\mathbb{Z}_d^n}\left[\sum_{j=1}^{t}\left(\sum_{(C,\alpha)\in E_j}\sum_{i=0}^{d-1}\left|\hat{Q}_\alpha^{f,i}\right|^2\right)\left(\sum_{(C,\alpha)\in E_j}\sum_{i=0}^{d-1}\left|h_{S_\alpha}^i(C)\right|^2\right)\right] \\
&\le \frac{\binom{k}{\ell}^2}{4}\mathbb{E}_{\sigma\sim\mathbb{Z}_d^n} \\
&\quad\left[\sum_{j=1}^{t}\left(\ell!\max_{j\le t,(C,\alpha)\in E_j}\sum_{i=0}^{d-1}\left|\hat{Q}_\alpha^{f,i}\right|^2\right)\left(\sum_{(C,\alpha)\in E_j}\sum_{i=0}^{d-1}\left|h_{S_\alpha}^i(C)\right|^2\right)\right] \\
&\le \frac{\binom{k}{\ell}^2}{4}\mathbb{E}_{\sigma\sim\mathbb{Z}_d^n} \\
&\quad\left[\left(\ell!\max_{j\le t,(C,\alpha)\in E_j}\sum_{i=0}^{d-1}\left|\hat{Q}_\alpha^{f,i}\right|^2\right)\left(\sum_{j=1}^{t}\sum_{(C,\alpha)\in E_j}\sum_{i=0}^{d-1}\left|h_{S_\alpha}^i(C)\right|^2\right)\right] \\
&\le \frac{\binom{k}{\ell}^2}{4}\mathbb{E}_{\sigma\sim\mathbb{Z}_d^n} \\
&\quad\left[\left(\ell!\max_{j\le t,(C,\alpha)\in E_j}\sum_{i=0}^{d-1}\left|\hat{Q}_\alpha^{f,i}\right|^2\right)\left(\max_{\alpha\in\mathbb{Z}_d^\ell}\sum_{C\in X_\ell}\sum_{i=0}^{d-1}\left|h_{S_\alpha}^i(C)\right|^2\right)\right] \\
&\le \frac{\binom{k}{\ell}^2}{4}\mathbb{E}_{\sigma\sim\mathbb{Z}_d^n} \\
&\quad\left[\left(\ell!\max_{j\le t,(C,\alpha)\in E_j}\sum_{i=0}^{d-1}\left|\hat{Q}_\alpha^{f,i}\right|^2\right)\left(|X_\ell|\max_{\alpha\in\mathbb{Z}_d^\ell}\sum_{i=0}^{d-1}\mathbb{E}_{C\sim X_\ell}\left[h_{S_\alpha}^i(C)^2\right]\right)\right] \\
&\le \frac{\binom{k}{\ell}^2 d\ell!\,|X_\ell|}{4}\mathbb{E}_{\sigma\sim\mathbb{Z}_d^n}\left[\left(\max_{j\le t,(C,\alpha)\in E_j}\sum_{i=0}^{d-1}\left|\hat{Q}_\alpha^{f,i}\right|^2\right)\right]\max_{\alpha\in\mathbb{Z}_d^\ell}\sum_{i=0}^{d-1}\|h_{S_\alpha}^i\|_2^2 \\
&\le \frac{\binom{k}{\ell}^2 d\ell!\,|X_\ell|}{4}\max_{\alpha\in\mathbb{Z}_d^\ell}\sum_{i=0}^{d-1}\|h_{S_\alpha}^i\|_2^2 \,.
\end{aligned}$$

Before we can apply Lemma 15 we must address a technicality. The range of $b = b_\ell$ might include complex numbers, but Lemma 15 only applies to functions b with range $\mathbb{R}$. For $c, d \in \mathbb{R}$ we adopt the notation $Im\left(c + d\sqrt{-1}\right) = d$ and $Re\left(c + d\sqrt{-1}\right) = c$. We observe that

$$\begin{aligned}
\mathbb{E}_{\sigma\sim\mathbb{Z}_d^n}\left[b(\sigma)\overline{b(\sigma)}\right] &= \mathbb{E}_{\sigma\sim\mathbb{Z}_d^n}\left[Re(b(\sigma))^2 + Im(b(\sigma))^2\right] \\
&= \|Re(b)\|_2^2 + \|Im(b)\|_2^2 \,.
\end{aligned}$$

We first observe that $Re(b)$ and $Im(b)$ are both degree ℓ functions because b is a degree ℓ function.

Now we can apply Lemma 15 to get

$$\begin{aligned}\mathbf{E}_{\sigma\sim\mathcal{D}'}\left[\left|Re\left(b\left(\sigma\right)\right)\right|\right] &\leq \frac{2\left(\ln d'/c_0\right)^{\ell/2}}{d^\ell}\|Re(b)\|_2 \\ &\leq \frac{2\left(\ln d'/c_0\right)^{\ell/2}}{d^\ell}\sqrt{\mathbb{E}_{\sigma\sim\mathbb{Z}_d^n}\left[b\left(\sigma\right)\overline{b\left(\sigma\right)}\right]} \\ &\leq \left(\binom{k}{\ell}\sqrt{\ell!d}\right)\left(\ln d'/c_0\right)^{\ell/2}\sqrt{|X_\ell|}\max_{\alpha\in\mathbb{Z}_d^\ell}\sum_{i=0}^{d}\|h_{S_\alpha}\left(\sigma\right)\|_2 \ .\end{aligned}$$

A symmetric argument can be used to bound $\mathbf{E}_{\sigma\sim\mathcal{D}'}\left[Im\left(b\left(\sigma\right)\right)\right]$. Now because

$$\left|b\left(\sigma\right)\right| \leq \left|Re\left(b\left(\sigma\right)\right)\right| + \left|Im\left(b\left(\sigma\right)\right)\right| \ ,$$

it follows that

$$\begin{aligned}\mathbf{E}_{\sigma\sim\mathcal{D}'}\left[\left|\frac{1}{|X_\ell|}b\left(\sigma\right)\right|\right] &\leq 2\left(\binom{k}{\ell}\sqrt{\ell!d}\right)\left(\frac{1}{|X_\ell|}\right)\left(\left(\ln d'/c_0\right)^{\ell/2}\right)\max_{\alpha\in\mathbb{Z}_d^\ell}\sum_{i=0}^{d-1}\|h_{S_\alpha}\left(\sigma\right)\|_2\sqrt{|X_\ell|} \\ &\leq 2\left(\binom{k}{\ell}\sqrt{\ell!d}\right)\left(\ln d'/c_0\right)^{\ell/2}\max_{\alpha\in\mathbb{Z}_d^\ell}\sum_{i=0}^{d-1}\frac{\|h_{S_\alpha}\left(\sigma\right)\|_2}{\sqrt{|X_\ell|}} \ .\end{aligned}$$

◀

We will use Fact 24 to prove Lemma 25. The proof of Fact 24 is found in [26, Lemma 7]. We include it here for completeness.

▶ **Fact 24.** *[26] If $h : X_k \times \mathbb{Z}_d \to \mathbb{R}$ satisfies $\|h\|_2^2 = 1$ then for any $i \in \mathbb{Z}_d$, $0 \leq \ell \leq k$ and $S \subseteq [k]$ of size $|S| = \ell$ we have $\sum_{i=0}^{d-1}\|h_S^i\|_2^2 \leq d$.*

Proof. First notice that for any C_ℓ, $S \subseteq [k]$ s.t $|S| = \ell$

$$\left|\{C \in X_k \,|\, C_{|S} = C_\ell\}\right| = \frac{|X_k|}{|X_\ell|} \ .$$

By applying the definition of h_ℓ along with the Cauchy-Schwartz inequality

$$\begin{aligned}\sum_{i=0}^{d-1}\|h_S^i\|_2^2 &= \sum_{i=0}^{d-1}\mathbb{E}_{C_\ell\sim X_\ell}\left[h_S^i\left(C_\ell\right)^2\right] \\ &= \left(\frac{|X_\ell|}{|X_k|}\right)^2\sum_{i=0}^{d-1}\mathbb{E}_{C_\ell\sim X_\ell}\left[\left(\sum_{C\in X_k, C_{|S}=C_\ell}h\left(C,i\right)\right)^2\right] \\ &\leq \left(\frac{|X_\ell|}{|X_k|}\right)^2\sum_{i=0}^{d-1}\mathbb{E}_{C_\ell\sim X_\ell}\left[\frac{|X_k|}{|X_\ell|}\left(\sum_{C\in X_k, C_{|S}=C_\ell}h\left(C,i\right)^2\right)\right] \\ &\leq \left(\frac{|X_\ell|}{|X_k|}\right)d\mathbb{E}_{C_\ell\sim X_\ell, i\sim\mathbb{Z}_d}\left[\left(\sum_{C\in X_k, C_{|S}=C_\ell}h\left(C,i\right)^2\right)\right] \\ &= d\mathbb{E}_{C\sim U_k}\left[h\left(C\right)^2\right] = d\|h\|_2^2 = d \ .\end{aligned}$$

◀

▶ **Lemma 25.** *Let $r = r(f)$, let $\mathcal{D}' \subseteq \{0,\ldots,d-1\}^n$ be a set of secret mappings and let $d' = d^n/|\mathcal{D}'|$. Then $\kappa_2(\mathcal{D}') = O_k\left(\left(\ln d'/n\right)^{r/2}\right)$*

Proof. Let $h : X_k \to \mathbb{R}$ be any function such that $\mathbb{E}_{U_k}\left[h^2\right] = 1$. Using Lemma 21 and the definition of r,

$$\begin{aligned} |\Delta(\sigma, h)| &= \left| \sum_{\ell=r}^{k} \frac{1}{|X_\ell|} b_\ell(\sigma) \right| \\ &\leq \sum_{\ell=r}^{k} \left| \frac{1}{|X_\ell|} b_\ell(\sigma) \right| . \end{aligned}$$

We apply Lemma 23 and Fact 24 to get

$$\begin{aligned} \mathbb{E}_{\sigma\sim\mathcal{D}'}\left[|\Delta(\sigma, h)|\right] &\leq 2\sum_{\ell=r}^{k} \binom{k}{\ell} \sqrt{\ell! d} \left(\ln d'/c_0\right)^{\ell/2} \max_{\alpha\in\mathbb{Z}_d} \sum_{i=0}^{d-1} \frac{\|h^i_{S_\alpha}(\sigma)\|_2}{\sqrt{|X_\ell|}} \\ &\leq \sum_{\ell=r}^{k} \left(\binom{k}{\ell} d\sqrt{\ell!} \right) \left(\frac{2\left(\ln d'/c_0\right)^{\ell/2}}{\sqrt{|X_\ell|}} \right) \\ &\leq O_k\left(\frac{\left(\ln d'\right)^{\ell/2}}{n^{r/2}} \right) . \end{aligned}$$

◀

▶ **Theorem 16** (restated). *There exists a constant $c_Q > 0$ such that for any $\epsilon > 1/\sqrt{n}$ and $q \geq n$ we have*

$$\mathrm{SDN}\left(\mathcal{Z}_{\epsilon,f}, \frac{c_Q \left(\log q\right)^{r/2}}{n^{r/2}}, 2e^{-n\cdot\epsilon^2/2} \right) \geq q ,$$

where $r = r(f)$ is the distributional complexity of f.

Proof of Theorem 16. First note that, by Chernoff bounds, for any solution $\tau \in \mathbb{Z}_d^n$ the fraction of assignments $\sigma \in \mathbb{Z}_d^n$ such that τ and σ are ϵ-correlated (e.g., $H(\sigma,\tau) \leq \frac{n(d-1)}{d} - \epsilon\cdot n$) is at most $e^{-2n\cdot\epsilon^2}$. In other words $|\mathcal{D}_\sigma| \geq \left(1 - e^{-2n\cdot\epsilon^2}\right)|\mathbb{Z}_d^n|$, where $\mathcal{D}_\sigma = \mathbb{Z}_d^n \setminus \left\{ \sigma' \,\middle|\, H(\sigma,\sigma') \leq \frac{n(d-1)}{d} - \epsilon\cdot n \right\}$ Let $\mathcal{D}' \subseteq \mathcal{D}_\sigma$ be a set of distributions of size $|\mathcal{D}_\sigma|/q$. Then for $d' = d^n/|\mathcal{D}'| = q\cdot d^n/|\mathcal{D}_\sigma|$, by Lemma 25 we get

$$\kappa_2(\mathcal{D}') = O_k\left(\frac{\left(\ln d'\right)^{r/2}}{n^{r/2}} \right) \tag{1}$$

$$= O_k\left(\frac{\left(\ln q\right)^{r/2}}{n^{r/2}} \right) , \tag{2}$$

where the last line follows by Sterling's Approximation

$$q = d'|\mathcal{D}_\sigma|/d^n = d'|\mathcal{D}_\sigma|/d^n \approx d'c'\sqrt{\frac{d}{n}}$$

for a constant c'. The claim now follows from the definition of SDN. ◀

The proof of Theorem 14 follows from Theorem 16 and the following result of Feldman et al. [26].

▶ **Theorem 13** (restated). *[26, Theorems 10 and 12] For $\kappa > 0$ and $\eta \in (0,1)$ let $d' = \mathrm{SDN}(\mathcal{Z}_{\epsilon,f}, \kappa, \eta)$ be the statistical dimension of the distributional search problem $\mathcal{Z}_{\epsilon,f}$. Any randomized statistical algorithm that, given access to a VSTAT$\left(\frac{1}{3\kappa^2}\right)$ oracle (resp. 1-MSTAT(L)) for the distribution Q^f_σ for a secret mapping σ chosen randomly and uniformly from $\mathbb{Z}_d^n$, succeeds in finding a mapping $\tau \in \mathbb{Z}_d^n$ that is ϵ-correlated with σ with probability $\Lambda > \eta$ over the choice of distribution and internal randomness requires at least $\frac{\Lambda-\eta}{1-\eta}d'$ (resp. $\Omega\left(\frac{1}{L}\min\left\{\frac{d'(\Lambda-\eta)}{1-\eta}, \frac{(\Lambda-\eta)^2}{\kappa^2}\right\}\right)$) calls to the oracle.*

▶ **Theorem 14** (restated). *Let $\sigma \in \mathbb{Z}_d^n$ denote a secret mapping chosen uniformly at random, let Q^f_σ be the distribution over $X_k \times \mathbb{Z}_d$ induced by a function $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$ with distributional complexity $r = r(f)$. Any randomized statistical algorithm that finds an assignment τ such that τ is $\left(\sqrt{\frac{-2\ln(\eta/2)}{n}}\right)$-correlated with σ with probability at least $\Lambda > \eta$ over the choice of σ and the internal randomness of the algorithm needs at least m calls to the 1-MSTAT(L) oracle (resp. VSTAT$\left(\frac{n^r}{2(\log n)^{2r}}\right)$ oracle) with $m \cdot L \geq c_1\left(\frac{n}{\log n}\right)^r$ (resp. $m \geq n^{c_1 \log n}$) for a constant $c_1 = \Omega_{k,1/(\Lambda-\eta)}(1)$ which depends only on the values k and $\Lambda - \eta$. In particular if we set $L = \left(\frac{n}{\log n}\right)^{r/2}$ then our algorithms needs at least $m \geq c_1\left(\frac{n}{\log n}\right)^{r/2}$ calls to 1-MSTAT(L).*

E Security Proofs

▶ **Theorem 19** (restated). *Let f be a function with evenly distributed output (Definition 17), let $\sigma \sim \mathbb{Z}_d^n$ denote the secret mapping, let $\epsilon > 0$ be any constant and suppose that for every $C \in X_k$ we are given labels $\ell_C \in \mathbb{Z}_d$ s.t. $\Pr_{C\sim X_k}[f(\sigma(C)) = \ell_C] \geq \frac{1}{d} + \epsilon$. There is a polynomial time algorithm (in n, m,$1/\epsilon$) that finds a mapping $\sigma' \in \mathbb{Z}_d^n$ such that σ' is $\epsilon/2$-correlated with σ with probability at least $\frac{\epsilon}{2d^2}$*

Proof of Theorem 19. Let $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$ be a function with evenly distributed output. We select fix $C^{-1} \sim X_{k-1}$ and $i \sim [n] \setminus C^{-1}$. Given $j \in [n] \setminus C^{-1}$ we let $C_j = (C^{-1}, j) \in X_k$ denote the corresponding clause. Now we generate the mapping σ' by selecting $\sigma'(i)$ at random, and setting $\sigma'(j) = \sigma'(i) + \ell_{C_j} - \ell_{C_i} \mod d$ for $j \in [n] \setminus C_i$. For $j \in C^{-1}$ we select $\sigma'(j)$ at random. We let $\mathbf{GOOD}\left(C^{-1}, i, \sigma'\right)$ denote the event that

$$\Pr_{j\sim[n]\setminus C^{-1}}\left[\ell_{C_j} = f\left(\sigma\left(C_j\right)\right)\right] \geq \frac{1}{d} + \epsilon/2 \ ,$$

$\sigma'(i) = \sigma(i)$ and $\ell_{C_i} = f(\sigma(C_i))$. Assume that the event $\mathbf{GOOD}\left(C^{-1}, i\right)$ occurs, in this case for each j s.t. $\ell_{C_j} = f(\sigma(C_j))$ we have

$$\begin{aligned}
\sigma'(j) - \sigma(j) &\equiv \left(\sigma'(i) + \ell_{C_j} - \ell_{C_i}\right) - \sigma(j) \mod d \\
&\equiv \left(\ell_{C_j} - \ell_{C_i}\right) + \sigma(i) - \sigma(j) \mod d \\
&\equiv g\left(\sigma\left(C^{-1}\right)\right) + \sigma(j) - \left(g\left(\sigma\left(C^{-1}\right)\right) + \sigma(i)\right) + \sigma(i) - \sigma(j) \mod d \\
&\equiv 0 \mod d \ .
\end{aligned}$$

Therefore, we have

$$\frac{n - H(\sigma, \sigma')}{n} \geq \frac{1}{d} + \epsilon/2 - \frac{k-1}{n} \ .$$

We note that by Markov's inequality the probability of success is at least

$$\Pr_{C^{-1}\sim X_{k-1}}\left[\mathbf{GOOD}\left(C^{-1}\right)\right] \geq \frac{\epsilon}{2d^2} \ . \qquad \blacktriangleleft$$

Before proving Theorem 18 we introduce some notation and prove an important claim. We use $\mathcal{A}_{C_1,\ldots,C_m} : (X_k)^\lambda \to \mathbb{Z}_d^\lambda$ to denote an adversary who sees the challenges $C_1,\ldots,C_m \in X_k$ and the corresponding responses $f(\sigma(C_1)),\ldots,f(\sigma(C_m))$. $\mathcal{A}_{C_1,\ldots,C_m}(C'_1,\ldots,C'_\lambda) \in \mathbb{Z}_d^\lambda$ denotes the adversaries prediction of $f(\sigma(C'_1)),\ldots,f(\sigma(C'_\lambda))$. Given a function $b : (X_k)^\lambda \to \mathbb{Z}_d^\lambda$, challenges $C'_1,\ldots,C'_\lambda \in X_k$ and responses $f(\sigma(C'_1)),\ldots,f(\sigma(C'_\lambda))$ we use $\mathcal{P}_{b,i,C'_1,\ldots,C'_m} : X_k \times [t] \to \mathbb{Z}_d \cup \{\perp\}$ to predict the value of a clause $C \in X_k$

$$\mathcal{P}_{b,C'_1,\ldots,C'_\lambda}(C,i) = \begin{cases} b\left(\hat{C}_1,\ldots,\hat{C}_\lambda\right)[i], & \text{if } f\left(\sigma\left(\hat{C}_j\right)\right) = b\left(\hat{C}_1,\ldots,\hat{C}_\lambda\right)[j] \ \ \forall j < i \\ \perp, & \text{otherwise} \end{cases}$$

where $\hat{C}_i = C$ and $\hat{C}_j = C'_j$ for $j \neq i$. We allow our predictor $\mathcal{P}_{b,C'_1,\ldots,C'_\lambda}(C,i)$ to output $\perp$ when it is unsure. Informally, Claim 26 says that for $b = \mathcal{A}_{C_1,\ldots,C_m}$ our predictor $\mathcal{P}_{b,i,C'_1,\ldots,C'_m}$ is reasonably accurate whenever it is not unsure. Briefly, Claim 26 follows because for $b = \mathcal{A}_{C_1,\ldots,C_m}$ we have

$$\Pr[\mathbf{Wins}(\mathcal{A},n,m,\lambda)] = \prod_{i=1}^{d} \Pr_{\substack{C\sim X_k \\ C_1,\ldots,C_m\sim X_k \\ C'_1,\ldots,C'_\lambda\sim X_k}} \left[\mathcal{P}_{b,C'_1,\ldots,C'_\lambda}(C,i) = f(\sigma(C)) \;\middle|\; \mathcal{P}_{b,C'_1,\ldots,C'_\lambda}(C,i) \neq \perp\right] .$$

▶ **Claim 26.** *Let $\mathcal{A}$ be an adversary s.t $\Pr[\mathbf{Wins}(\mathcal{A},n,m,\lambda)] > \left(\frac{1}{d}+\epsilon\right)^\lambda$ and let $b = \mathcal{A}_{C_1,\ldots,C_m}$ then*

$$\Pr_{\substack{i\sim[\lambda],C\sim X_k \\ C_1,\ldots,C_m\sim X_k \\ C'_1,\ldots,C'_\lambda\sim X_k}} \left[\mathcal{P}_{b,C'_1,\ldots,C'_\lambda}(C,i) = f(\sigma(C)) \;\middle|\; \mathcal{P}_{b,C'_1,\ldots,C'_\lambda}(C,i) \neq \perp\right] \geq \left(\frac{1}{d}+\epsilon\right) .$$

Proof of Claim 26. We draw examples $(C_1, f(\sigma(C_1))),\ldots,(C_m, f(\sigma(C_m)))$ to construct $b = \mathcal{A}_{C_1,\ldots,C_m}$. Given a random length-$\lambda$ password challenge $(C'_1,\ldots,C'_\lambda) \in (X_k)^t$ we let

$$p_j = \Pr_{C,C_1,\ldots,C_m,C'_1,\ldots,C'_\lambda\sim X_k} \left[\mathcal{P}_{b,j,C'_1,\ldots,C'_t}(C) = f(\sigma(C)) \;\middle|\; \mathcal{P}_{b,j,C'_1,\ldots,C'_\lambda}(C) \neq \perp\right]$$

denote the probability that the adversary correctly guesses the response to the j'th challenge conditioned on the event that the adversary correctly guesses all of the earlier challenges. Observe that

$$\Pr_{C,C_1,\ldots,C_m,C'_1,\ldots,C'_{\lambda-1}\sim X_k,i\sim[t]} \left[\mathcal{P}_{b,i,C'_1,\ldots,C'_\lambda}(C,i) = f(\sigma(C))\right] = \sum_{i=1}^{\lambda} p_i/\lambda ,$$

so it suffices to show that $\sum_{i=1}^{\lambda} p_i/\lambda \geq \frac{1}{d}+\epsilon$. We obtain the following constraint

$$\begin{aligned} \prod_{i=1}^{\lambda} p_i &= \prod_{i=1}^{\lambda} \Pr_{C,C_1,\ldots,C_m,C'_1,\ldots,C'_\lambda\sim X_k} \left[\mathcal{P}_{b,j,C'_1,\ldots,C'_\lambda}(C) = f(\sigma(C)) \;\middle|\; \mathcal{P}_{b,j,C'_1,\ldots,C'_\lambda}(C) \neq \perp\right] \\ &= \prod_{i=1}^{\lambda} \Pr_{C_1,\ldots,C_m,C'_1,\ldots,C'_\lambda\sim X_k} \\ &\quad \left[\mathcal{A}_{C_1,\ldots,C_m}\left(C'_1,\ldots,C'_\lambda\right)[i] = f\left(\sigma\left(C'_i\right)\right) \;\middle|\; \forall j < i.\ \mathcal{A}_{C_1,\ldots,C_m}\left(C'_1,\ldots,C'_\lambda\right)[j] = f\left(\sigma\left(C'_j\right)\right)\right] \\ &= \Pr_{C_1,\ldots,C_m,C'_1,\ldots,C'_\lambda\sim X_k} \left[\mathcal{A}_{C_1,\ldots,C_m}\left(C'_1,\ldots,C'_\lambda\right) = \left(f\left(\sigma\left(C'_1\right)\right),\ldots,f\left(\sigma\left(C'_\lambda\right)\right)\right)\right] \\ &\geq \left(\frac{1}{d}+\epsilon\right)^\lambda . \end{aligned}$$

If we minimize $\sum_{i=1}^{t} p_i/\lambda$ subject to the constraint $\prod_{i=1}^{\lambda} p_i \geq \left(\frac{1}{d}+\epsilon\right)^{\lambda}$ then we obtain the desired upper bound $\sum_{i=1}^{\lambda} p_i/\lambda \geq \frac{1}{d}+\epsilon$. ◀

▶ **Theorem 18** (restated). *Suppose that f has evenly distributed output, but that f is not $\mathbf{UF-RCA}(n,m,\lambda,\delta)$ − secure for $\delta > \left(\frac{1}{d}+\epsilon\right)^{\lambda}$. Then there is a probabilistic polynomial time algorithm (in n, m, λ and $1/\epsilon$) that extracts a string $\sigma' \in \mathbb{Z}_d^n$ that is $\epsilon/8$-correlated with σ with probability at least $\frac{\epsilon^3}{(8d)^2}$ after seeing $m+\lambda$ example challenge response pairs.*

Proof of Theorem 18. Given random clauses $C_1,\ldots,C_m,C'_1,\ldots,C'_\lambda \sim X_k$ we let $\mathbf{Good}(C_1,\ldots,C_m,C'_1,\ldots,C'_\lambda)$ denote the event that

$$\Pr_{i\sim[t],C\sim X_k}\left[\mathcal{P}_{b,C'_1,\ldots,C'_\lambda}(C,i) = f(\sigma(C)) \;\middle|\; \mathcal{P}_{b,C'_1,\ldots,C'_\lambda}(C,i) \neq \perp\right] \geq \left(\frac{1}{d}+\frac{\epsilon}{2}\right).$$

By Markov's Inequality and Claim 26 we have $\Pr[\mathbf{Good}(C_1,\ldots,C_m,C'_1,\ldots,C'_\lambda)] \geq \frac{\epsilon}{2}$. Here, $b = \mathcal{A}_{C_1,\ldots,C_m}$ and

$$\mathcal{P}_{b,C'_1,\ldots,C'_\lambda}(C,i) = \begin{cases} b\left(\hat{C}_1,\ldots,\hat{C}_\lambda\right)[i], & \text{if } f\left(\sigma\left(\hat{C}_j\right)\right) = b\left(\hat{C}_1,\ldots,\hat{C}_\lambda\right)[j] \;\; \forall j<i \\ \perp, & \text{otherwise} \end{cases}$$

Assuming that the event $\mathbf{Good}(C_1,\ldots,C_m,C'_1,\ldots,C'_\lambda)$ occurs we obtain labels for each clause $C \in X_k$ by selecting a random permutation $\pi:[\lambda]\to[\lambda]$, setting $i=1$ and setting $\ell_C = \mathcal{P}_{b,C'_1,\ldots,C'_\lambda}(C,\pi(i))$ – if $\ell_C \neq \perp$ then we increment i and repeat. Note that we will always find a label $\ell_C \neq \perp$ within t attempts because $\mathcal{P}_{b,C'_1,\ldots,C'_\lambda}(C,1) \neq \perp$. Let **GoodLabels** denote the event that

$$\Pr_{C\sim X_k}[G_C] \geq \frac{1}{d}+\frac{\epsilon}{4},$$

where G_C is the indicator random variable for the event $\ell_C = f(\sigma(C))$. We have

$$\mathbb{E}\left[\frac{1}{|X_k|}\sum_{C\in X_k} G_C\right] \geq \frac{1}{d}+\frac{\epsilon}{2},$$

so we can invoke Markov's inequality again to argue that $\Pr[\mathbf{GoodLabels} \mid \mathbf{Good}] \geq \frac{\epsilon}{4}$. If the event **GoodLabels** occurs then we can invoke Theorem 19 to obtain σ' that is $\epsilon/8$-correlated with σ with probability at least $\frac{\epsilon}{8d^2}$. Our overall probability of success is

$$\frac{\epsilon}{8d^2}\times\frac{\epsilon}{4}\times\frac{\epsilon}{2} = \frac{\epsilon^3}{(8d)^2}.$$ ◀

F Security Parameters of f_{k_1,k_2}

▶ **Claim 9** (restated). *Let $0 \leq k_1$ and $k_2 > 0$ be given and let $f = f_{k_1,k_2}$ we have $g(f) = \min\{k_1,10\}$, $r(f) = k_2+1$ and $s(f) = \min\left\{\frac{k_2+1}{2}, k_1+1, 11\right\}$.*

Proof of Claim 9. Let $f(x) = f_{k_1,k_2}(x) = x_{\left(\sum_{i=10}^{9+k_1} x_i \mod 10\right)} + \sum_{i=10+k_1}^{9+k_1+k_2} x_i \mod 10$. We first observe that if we fix the values of $x_{10},\ldots,x_{9+k_1} \in \mathbb{Z}_{10}$ and let $i' = \sum_{i=10}^{9+k_1} x_i \mod 10$ then $f'(x_0,\ldots,x_9,x_{10+k_1},\ldots,x_{9+k_1+k_2}) = x_{i'} + \sum_{i=10+k_1}^{9+k_1+k_2} x_i \mod 10$ is a linear function. Similarly, if we fix the values of $x_0 = \ldots = x_9 = c$ then the resulting function

$f'(x_{10},\ldots,x_{9+k_1+k_2}) = c + \sum_{i=10+k_1}^{9+k_1+k_2} x_i \mod 10$ is linear. Thus, $g(f) \leq \min\{10,k_1\}$. Now suppose that we don't fix all of the values $x_{10},\ldots,x_{9+k_1} \in \mathbb{Z}_{10}$ and at least one of the variables $x_0,\ldots,x_9$ is not fixed. In this case the resulting function will not be linear. Thus, $g(f) \geq \min\{k_1,10\}$. We also note that for any $\alpha \in \mathbb{Z}_{10}^{10+k_1+k_2}$ s.t. $H(\alpha) \leq k_2$ and $i,t \in \mathbb{Z}_{10}$ that

$$\Pr_{x\sim\mathbb{Z}_{10}^{10+k_1+k_2}}[f(x)=t \mid \alpha\cdot x \equiv i \mod 10] = \Pr_{x\sim\mathbb{Z}_{10}^{10+k_1+k_2}}[f(x)=t] = \frac{1}{10}\ .$$

Therefore,

$$\begin{aligned}
\hat{Q}_\alpha^{f,t} &= \mathbb{E}_{x\sim\mathbb{Z}_{10}^{10+k_1+k_2}}\left[Q^{f,t}(x)\,\chi_\alpha(x)\right] \\
&= \sum_{i=0}^{9} \Pr[\alpha\cdot x \equiv i \mod 10]\,\mathbb{E}_{x\sim\mathbb{Z}_{10}^{10+k_1+k_2}}\left[Q^{f,t}(x)\,\chi_\alpha(x)\;\middle|\;\alpha\cdot x\equiv i \mod 10\right] \\
&= \sum_{i=0}^{9} \exp\left(\frac{-2\pi i\sqrt{-1}}{10}\right)\Pr[\alpha\cdot x \equiv i \mod 10]\,\mathbb{E}_{x\sim\mathbb{Z}_{10}^{10+k_1+k_2}}\left[Q^{f,t}(x)\;\middle|\;\alpha\cdot x\equiv i \mod 10\right] \\
&= \frac{1}{10}\sum_{i=0}^{9} \exp\left(\frac{-2\pi i\sqrt{-1}}{10}\right)\mathbb{E}_{x\sim\mathbb{Z}_{10}^{10+k_1+k_2}}\left[Q^{f,t}(x)\;\middle|\;\alpha\cdot x\equiv i \mod 10\right] \\
&= 0\ ,
\end{aligned}$$

which implies that $r(f) \geq k_2+1$. Similarly, if we set $\alpha = (\alpha_0,\ldots,\alpha_{9+k_1+k_2})$ such that $\alpha_0 = 1$ and $\alpha_{10+k_1} = \ldots = \alpha_{9+k_1+k_2} = 1$ so that α has hamming weight k_2+1 then we can verify that $\hat{Q}_\alpha^{f,t} \neq 0$. ◀

G Security Upper Bounds

G.1 Statistical Algorithms

Theorem 27 demonstrates that our lower bound for statistical algorithms are asymptotically tight for our human computable password schemes f_{k_1,k_2}. In particular, we demonstrate that $m = \tilde{O}\left(n^{(k_2+1)/2}\right)$ queries to 1-MSTAT are sufficient for a statistical algorithm to recover σ.

▶ **Theorem 27.** *For $f = f_{k_1,k_2}$ there is a randomized algorithm that makes $O\left(n^{\max\{1,(k_2+1)/2\}}\log^2 n\right)$ calls to the 1-MSTAT$\left(n^{\lceil r(f_i)/2\rceil}\right)$ oracle and returns σ with probability $1-o(1)$.*

For binary functions $f' : \{0,1\}^k \to \{0,1\}$, Feldman et al. [26] gave a randomized statistical algorithm to find $\sigma' \in \{0,1\}^n$ using just $O\left(n^{r(f)/2}\log^2 n\right)$ calls to the 1-MSTAT $\left(n^{\lceil r(f)/2\rceil}\right)$ oracle. Their main technique is a discrete spectral iteration procedure to find the eigenvector (singular vector) with the largest eigenvalue (singular value) of a matrix M sampled from a distribution $M_{\sigma',p}$ over $\left|X_{\lfloor r(f)/2\rfloor}\right| \times \left|X_{\lceil r(f)/2\rceil}\right|$ matrices. With probability $1-o(1)$ this eigenvector will encode the value $\sum_{i\in C}\sigma'(i) \mod 2$ for each clause $C \in X_{r(f)/2}$. We show that the discrete spectral iteration algorithm of Feldman et al [26] can be extended to recover $\sigma \in \mathbb{Z}_{10}$ when f_{k_1,k_2} is one of our candidate human computable functions.

Discussion

We note that Theorem 27 cannot be extended to arbitrary functions $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$. Consider for example the unique function $f : \mathbb{Z}_{10}^6 \to \mathbb{Z}_{10}$ s.t. $f(x_1,\ldots,x_6) \equiv f'(x_1 \mod 2,\ldots,x_6 \mod 2) \mod 2$ and $f(x_1,\ldots,x_6) \equiv f''(x_1 \mod 5,\ldots,x_6 \mod 5) \mod 5$, where $f' : \mathbb{Z}_2^6 \to \mathbb{Z}_2$ and

$f'' : \mathbb{Z}_5^6 \to \mathbb{Z}_5$. By the Chinese Remainder Theorem instead of picking a secret mapping $\sigma \in \mathbb{Z}_{10}^n$ we could equivalently pick the unique secret mappings $\sigma_1 \in \mathbb{Z}_2^n$ and $\sigma_2 \in \mathbb{Z}_5^n$ s.t $\sigma \equiv \sigma_1 \mod 2$ and $\sigma \equiv \sigma_2 \mod 5$. Now drawing challenge response pairs from the distributions Q_σ^f is equivalent to drawing challenge-response pairs from the distributions $Q_{\sigma_1}^{f'}$ and $Q_{\sigma_2}^{f''}$. Suppose that $f'(x_1, \ldots, x_6) = x_1x_2 + x_3 + x_4 + x_5 + x_6 \mod 2$, and $f''(x_1, \ldots, x_6) = x_1$. Then we have $r(f) = \min(r(f'), r(f'')) = r(f'') = 1$, but $r(f') = 4$. We can find σ_2 using $O\left(n \log^2 n\right)$ calls to 1-MSTAT(n), but to find σ we must also recover σ_1. This provably requires at least $\tilde{\Omega}\left(n^{r(f')/2}\right) = \tilde{\Omega}\left(n^2\right)$ calls to 1-MSTAT$\left(n^2\right)$.

Background

The proof of Theorem 27 relies on the discrete spectral iteration algorithm of [26]. We begin by providing a brief overview of their algorithm. In their setting the secret mapping σ is defined over the binary alphabet $\mathbb{Z}_2^n$. Let $c_1 = \lceil \frac{r(f)}{2} \rceil, c_2 = \lfloor \frac{r(f)}{2} \rfloor$ and let $\delta \in [0, 2] \backslash \{1\}$. They use σ to define a distribution over $|X_{c_1}| \times |X_{c_2}|$ matrices $M_{\sigma,\delta,p} = \hat{M}\left(Q_{\sigma,\delta,p}\right) - Jp$, where J denotes the all ones matrix. For $(C_1) \in X_{c_1}$, $(C_2) \in X_{c_2}$ such that $C_1 \bigcap C_2 = \emptyset$ we have

$$\hat{M}\left(Q_{\sigma,\delta,p}\right)[(C_1), (C_2)] = \begin{cases} 1, & \text{with probability } (p(2-\delta)) \text{ if } \sum_{j \in C_1 \cup C_2} \sigma(j) \equiv 0 \mod 2 \\ 1, & \text{with probability } (p\delta) \text{ if } \sum_{j \in C_1 \cup C_2} \sigma(j) \not\equiv 0 \mod 2 \\ 0, & \text{otherwise} \end{cases}.$$

Given a vector $x \in \{\pm 1\}^{|X_{c_2}|}$ (resp. $y \in \{\pm 1\}^{|X_{c_1}|}$) $M_{\sigma,\delta,p}x$ defines a distribution over vectors in $\mathbb{R}^{|X_{c_1}|}$ (resp. $M_{\sigma,p}^T y$ defines a distribution over vectors in $\mathbb{R}^{|X_{c_1}|}$).

If $r(f)$ is even then the largest eigenvalue of $\mathbb{E}[M_{\sigma,\delta,p}]$ has a corresponding eigenvector $x^* \in \{\pm 1\}^{X_{r(f)/2}}$, where for $C_i \in X_{r(f)/2}$ we have $x^*[C_i] = 1$ if $\sum_{j \in C_i} \sigma(j) \equiv 1 \mod 2$; otherwise $x^*[C_i] = -1$ (if $r(f)$ is odd then we consider the top singular value instead). Feldman et al [26] use discrete spectral iteration to find x^* (or $-x^*$). Given x^* it is easy to find σ using Gaussian Elimination.
The discrete spectral iteration algorithm of Feldman et al [26] starts with a random vector $x^0 \in \{0,1\}^{|X_{c_2}|}$. They then sample $x^{i+1} \sim M_{\sigma,p}x^i$ and execute a normalization step to ensure that $x^{i+1} \in \{0,1\}^{|X_{k_2}|}$. When $r(f)$ is odd, power iteration has two steps: draw a sample $y^i \sim M_{\sigma,\delta,p}x^i$ and sample from the distribution $x^{i+1} = M_{\sigma,\delta,p}^T y^i$. They showed that $O\left(\log |X_{r(f)}|\right)$ iterations suffice to recover σ whenever $p = \frac{K \log |X_{r(f)}|}{(\delta-1)^2 \sqrt{|X_{r(f)}|}}$, and that for a vector $x \in \{0,1\}^{|X_{k_2}|}$ (resp. $y \in \{\pm 1\}^{|X_{k_1}|}$) it is possible to sample from $M_{\sigma,\delta,p}x$ (resp. $M_{\sigma,\delta,p}^T y$) using $O(1/p)$ queries to 1-MSTAT$(|X_{c_1}|)$.

Our Reduction

The proof of Theorem 27 uses a reduction to the algorithm of Feldman et al[26].

Proof of Theorem 27 (sketch). Given a mapping $\sigma \in \mathbb{Z}_d^n$ and a number $i \in \mathbb{Z}_d$ we define a mapping $\sigma_i \in \mathbb{Z}_2^n$ where

$$\sigma_i(j) = \begin{cases} 1, & \textbf{if} \sigma(j) = i \\ 0, & \text{otherwise} \end{cases}.$$

Clearly, to recover σ it is sufficient to recover σ_i for each $i \in \mathbb{Z}_d$. Therefore, to prove Theorem 27 it suffices to show that given $x \in \{\pm 1\}^{|X_{k_2}|}$ (resp. $y \in \{\pm 1\}^{|X_{k_1}|}$) we can sample from

the distribution $M_{\sigma_i,\delta,p}x$ (resp. $M^T_{\sigma_i,\delta,p}y$) using $O(1/p)$ queries to 1-MSTAT $\left(\left|X_{\lceil r(f)/2\rceil}\right|\right)$ for each $i \in \{0,\ldots,d-1\}$, where 1-MSTAT uses the distribution Q^f_σ. In general, this will not possible for arbitrary functions f. However, Lemma 28 shows that for our candidate human computable functions $f_{1,3}, f_{2,2}$ we can sample from the distributions $M_{\sigma_i,\delta,p}x$ (resp. $M^T_{\sigma_i,\delta,p}y$). The proof of Lemma 28 is similar to the proof of [26, Lemma 10]. ◀

▶ **Lemma 28.** *Given vectors $\vec{x} \in \{\pm 1\}^{|X_{c_1}|}, \vec{y} \in \{\pm 1\}^{|X_{c_2}|}$ we can sample from $M_{\sigma_i,\delta,p}x$ and $M^T_{\sigma,\delta,p}y$ using $O\left(n^{(k_2+1)/2}\log^2 n\right)$ calls to the 1-MSTAT$\left(n^{\lceil r(f)/2\rceil}\right)$ oracle for $f = f_{k_1,k_2}$.*

The proof of Lemma 28 relies on Fact 29.

▶ **Fact 29.** *For each $j,t \in \mathbb{Z}_{10}$ we have*

$$\Pr_{(x_0,\ldots,x_{9+k_1+k_2})\sim\mathbb{Z}_{10}^{10+k_1+k_2}}\left[x_t + \sum_{i=10+k_1}^{10+k_1+k_2} x_i \equiv j \;\middle|\; f_{k_1,k_2}(x_0,\ldots,x_{9+k_1+k_2}) \equiv j \mod 10\right] = \frac{\left(\frac{9}{10}\left(\frac{1}{10}\right)+\frac{1}{10}\right)\left(\frac{1}{10}\right)}{\left(\frac{1}{10}\right)} = \frac{19}{100}\ ,$$

and

$$\Pr_{(x_0,\ldots,x_{9+k_1+k_2})\sim\mathbb{Z}_{10}^{10+k_1+k_2}}\left[x_t + \sum_{i=10+k_1}^{10+k_1+k_2} x_i \equiv j \;\middle|\; f_{k_1,k_2}(\sigma(x_0,\ldots,x_{9+k_1+k_2})) \not\equiv j \mod 10\right] = \frac{\left(\frac{9}{10}\left(\frac{1}{10}\right)+\frac{1}{10}(0)\right)\left(\frac{1}{10}\right)}{\left(\frac{1}{10}\right)} = \frac{9}{100}\ .$$

Proof of Lemma 28. Given a value $j \in \mathbb{Z}_{10}$ and a value $i \in \mathbb{Z}_{10}$ we let $x^i_j \in \{0,1\}$ be the indicator variable for the event $x_j = i$. By Fact 29 it follows that

$$\begin{aligned}
&\Pr_{(x_0,\ldots,x_{k_1+k_2+9})\sim\mathbb{Z}_{10}^{k_1+k_2+10}}\left[x^i_0 + x^i_{9+k_1} + \ldots + x^i_{7+k_1+c_1} \equiv 1 \mod 2 \;\middle|\; f_{k_1,k_2}(\sigma(x_0,\ldots,x_{9+k_1+k_2})) \equiv ic_1 \mod 10\right] \\
\neq\ &\Pr_{(x_0,\ldots,x_{k_1+k_2+9})\sim\mathbb{Z}_{10}^{k_1+k_2+10}}\left[x^i_0 + x^i_{9+k_1} + \ldots + x^i_{7+k_1+c_1} \equiv 1 \mod 2 \;\middle|\; f_{k_1,k_2}(\sigma(x_0,\ldots,x_{9+k_1+k_2})) \not\equiv ic_1 \mod 10\right]\ .
\end{aligned}$$

Now for f_{k_1,k_2} we define the function $h^{i,+} : X_{k_1+k_2+10} \times \mathbb{Z}_{10} \to X_{c_1} \cup \{\perp\}$ as follows

$$h^{i,+}(x_0,\ldots,x_{9+k_1+k_2}, f_{k_1,k_2}(\sigma(x_0,\ldots,x_{9+k_1+k_2}))) = \begin{cases} (x_0, x_{9+k_1},\ldots,x_{7+k_1+c_1}) & \text{if} f_{k_1,k_2}(\sigma(x_0,\ldots,x_{13})) \equiv ic_1 \mod 10 \\ \perp & \text{otherwise.} \end{cases}.$$

Intuitively, given a clause $C_1 \in X_{c_1}$ the probability that $h^{i,+}$ returns C_1 is greater if $\sum_{j\in C_1}\sigma_i(j) \equiv 1 \mod 2$.

Given a vector $x \in \{\pm 1\}^{|X_{c_1}|}$ we query our 1-MSTAT $(|X_{c_1}|+1)$ oracle $\lceil 10/p\rceil$ times with the function $h^{i,+}$ to sample from $M_{\sigma_i,\delta,p}x$. Let $q_1,\ldots,q_{\lceil 10/p\rceil} \in X_{c_1}$ denote the responses and let $x[q_j]$ denote the value of the vector x at index q_j. We observe that for some $\delta \neq 1$ we have

$$M_{\sigma_i,\delta,p}x[C] \sim \sum_{\substack{j\in\lceil 10/p\rceil \\ q_j\neq\perp}} x[q_j] - p\sum_{C'\in X_{c_1}} x[C']\ ,$$

for every $C \in X_{c_2}$. ◀

Algorithm 4: GaussianAttack

input : Clauses $C_1, \ldots, C_m \sim X_k$, and labels $f(\sigma(C_1)), \ldots, f(\sigma(C_m))$.
forall $S \in X_{g(f)}$, $\alpha \in \mathbb{Z}_d^{g(f)}$ **do**
  $\mathbf{LC} \leftarrow \emptyset$;
  // **LC** is the set of linear constraints extracted
  forall $C \in \{C_1, \ldots, C_m\}$ **do**
    $\mathbf{LC} \leftarrow \mathbf{LC} \bigcup \mathbf{TryExtract}(C, f(\sigma(C)), S, \alpha)$;
    if $|\mathbf{LC}| \geq n$ **then**
      $\sigma' \leftarrow \mathbf{LinearSolve}(LC)$;
      if $\forall i \in [m].\ f(\sigma'(C_i)) = f(\sigma(C_i)) \in C$ **then**
        return σ'
      end
    end
  end
end

Open Question

Can we precisely characterize the functions $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$ for which we can efficiently recover σ after seeing $\tilde{O}\left(n^{r(f)/2}\right)$ challenge-response pairs? Feldman et al. [26] gave a statistical algorithm that recovers the secret mapping whenever $d = 2$ after making $\tilde{O}\left(n^{r(f)/2}\right)$ queries to 1-MSTAT $\left(n^{r(f)/2}\right)$. While we show that the same algorithm can be used to recover σ after making $\tilde{O}\left(n^{r(f)/2}\right)$ queries to 1-MSTAT $\left(n^{r(f)/2}\right)$ in our candidate human computable password schemes with $d = 10$, we also showed that these results do not extend to all functions $f : \mathbb{Z}_d^k \to \mathbb{Z}_d$.

G.2 Gaussian Elimination

Most known algorithmic techniques can be modeled within the statistical query framework. Gaussian Elimination is a notable exception. As an example consider the function $f(x_1, \ldots, x_7) = x_1 + \ldots + x_7 \mod 10$ (in this example $r(f) = 7$ and $g(f) = 0$). Our previous results imply that any statistical algorithm would need to see at least $m = \tilde{\Omega}\left(n^{7/2}\right)$ challenge response pairs $(C, f(\sigma(C)))$ to recover σ. However, it is trivial to recover σ from $O(n)$ random challenge response pairs using Gaussian Elimination. In general, consider the following attacker shown in algorithm 4, which uses Gaussian Elimination. Algorithm 4 relies on the subroutine $\mathbf{TryExtract}(C, f(\sigma(C)), S, \alpha)$, which attempts to extract a linear constraint from $(C, f(\sigma(C)))$ under the assumption that $\sigma(S) = \alpha$. We assume $\mathbf{TryExtract}(C, f(\sigma(C)), S, \alpha)$ returns $\emptyset$ if it cannot extract a linear constraint. For example, if we assume that $\sigma(1) = 4$ and $\sigma(2) = 8$ and let $C = (i_0, i_1, ..., i_9, 1, 2, i_{10}, i_{11})$ (with $i_j \in [n] \setminus \{1, 2\}$) then we have $f_{2,2}(\sigma(C)) = \sigma(i_{4+8 \mod 10}) + \sigma(i_{10}) + \sigma(i_{11}) \mod 10$. In this case, $\mathbf{TryExtract}(C, f(\sigma(C)), \{1, 2\}, \{4, 8\})$ would return the constraint $f(\sigma(C)) = \sigma(i_2) + \sigma(i_{10}) + \sigma(i_{11}) \mod 10$. However, $\mathbf{TryExtract}(C, f(\sigma(C)), \{i_0, 2\}, \{4, 8\})$ would return $\emptyset$.

Fact 30 says that an attacker needs at least $m = \tilde{\Omega}\left(n^{1+g(f)}\right)$ challenge-response pairs to recover σ using Gaussian Elimination. This is because the probability that $\mathbf{TryExtract}(C, f(\sigma(C)) S, \alpha)$ extracts a linear constraint is at most $O\left(\left(\frac{|S|}{n}\right)^{-g(f)}\right)$, which is $O\left(n^{-g(f)}\right)$ for $|S|$ constant. The adversary needs $O(n)$ linearly independent constraints

to run Gaussian Elimination. If the adversary can see at most $\tilde{O}\left(n^{s(f)}\right)$ examples neither approach (Statistical Algorithms or Gaussian Elimination) can be used to recover σ.

▶ **Fact 30.** *Algorithm 4 needs to see at least* $m = \tilde{\Omega}\left(n^{1+g(f)}\right)$ *challenge-response pairs to recover* σ*.*

Remark G.2 explores the tradeoff between the adversary's running time and the number of challenge-response pairs that an adversary would need to see to recover σ using Gaussian elimination. In particular the adversary can recover σ from $\tilde{O}\left(n^{1+g(f)/2}\right)$ challenge-response pairs if he is willing to increase his running time by a factor of $d^{\sqrt{n}}$. In practice, this attack may be reasonable for $n \leq 100$ and $d = 10$, which means that it may be beneficial to look for candidate human computable functions f that maximize $\min\{r(f)/2, 1 + g(f)/2\}$ instead of $s(f)$ whenever $n \leq 100$.

▶ Remark. If the adversary correctly guesses value of $\sigma(S)$ for $|S| = n^{\epsilon}$ then he may be able to extract a linear constraint from a random example with probability $\Omega(1/n^{(1-\epsilon)g(f)})$. The adversary would only need $\tilde{O}\left(n^{1+(1-\epsilon)g(f)}\right)$ examples to solve for σ, but his running time would be proportional to $d^{\epsilon n}$ – the expected number of guesses before he is correct.

H Rehearsal Model

In this section we review the usability model of Blocki et al. [13]. Their usability model estimates the 'extra effort' that a user needs to expend to memorize and rehearse all of his secrets for a password management scheme. In this section we use $(\hat{c}, \hat{a})$ to denote a cue-association pair, and we use the variable t to denote time (days). In our context $(\hat{c}, \hat{a})$ might denote the association between a letter (e.g., 'e') and the secret digit associated with that letter (e.g., $\sigma(e)$). If the user does not rehearse an association $(\hat{c}, \hat{a})$ frequently enough then the user might forget it. Their are two main components to their usability model: rehearsal requirements and visitation schedules. Rehearsal requirements specify how frequently a cue-association pair must be used for a user to remember the association. Visitation schedules specify how frequently the user authenticates to each of his accounts and rehearses any cue-association pairs that are linked with the account.

H.1 Rehearsal Requirements

Blocki et al. [13] introduce a rehearsal schedule to ensure that the user remembers each cue-association pair.

▶ **Definition 31.** [13] A rehearsal schedule for a cue-association pair $(\hat{c}, \hat{a})$ is a sequence of times $t_0^{\hat{c}} < t_1^{\hat{c}} < ....$ For each $i \geq 0$ we have a *rehearsal requirement*, the cue-association pair must be rehearsed at least once during the time window $\left[t_i^{\hat{c}}, t_{i+1}^{\hat{c}}\right) = \{x \in \mathbb{R} \,\middle|\, t_i^{\hat{c}} \leq x < t_{i+1}^{\hat{c}}\}$.

A rehearsal schedule is *sufficient* if a user can maintain the association $(\hat{c}, \hat{a})$ by following the rehearsal schedule. The length of each interval $\left[t_i^{\hat{c}}, t_{i+1}^{\hat{c}}\right)$ may depend on the strength of the mnemonic techniques used to memorize and rehearse a cue-association pair $(\hat{c}, \hat{a})$ as well as i – the number of prior rehearsals [54, 52].

Expanding Rehearsal Assumption [13]: The rehearsal schedule given by $t_i^{\hat{c}} = 2^{i\mathbf{s}}$ is sufficient to maintain the association $(\hat{c}, \hat{a})$, where $\mathbf{s} > 0$ is a constant.

Table 6 Visitation Schedules - number of accounts visited with frequency λ (visits/days)

Schedule | λ	$\frac{1}{1}$	$\frac{1}{3}$	$\frac{1}{7}$	$\frac{1}{31}$	$\frac{1}{365}$
Very Active	10	10	10	10	35
Typical	5	10	10	10	40
Occasional	2	10	20	20	23
Infrequent	0	2	5	10	58

H.2 Visitation Schedules

Suppose that the user has m accounts $A_1, \ldots, A_m$. A visitation schedule for an account A_i is a sequence of real numbers $\tau_0^i < \tau_1^i < \ldots$, which represent the times when the account A_i is visited by the user. Blocki et al. [13] do not assume that the exact visitation schedules are known a priori. Instead they model visitation schedules using a random process with a known parameter λ_i based on $E\left[\tau_{j+1}^i - \tau_j^i\right]$ – the average time between consecutive visits to account A_i.

A rehearsal requirement $\left[t_i^{\hat{c}}, t_{i+1}^{\hat{c}}\right)$ can be satisfied naturally if the user visits a site A_j that uses the cue $\hat{c}$ ($\hat{c} \in c_j$) during the given time window. Here, c_j denote the set of cue-association pairs that the user must remember when logging into account A_j. Formally,

▶ **Definition 32.** [13] We say that a rehearsal requirement $\left[t_i^{\hat{c}}, t_{i+1}^{\hat{c}}\right)$ is *naturally satisfied* by a visitation schedule $\tau_0^i < \tau_1^i < \ldots$ if $\exists j \in [m], k \in \mathbb{N}$ s.t $\hat{c} \in c_j$ and $\tau_k^j \in \left[t_i^{\hat{c}}, t_{i+1}^{\hat{c}}\right)$. We use

$$ER_{t,\hat{c}} = \left|\left\{i \;\middle|\; t_{i+1}^{\hat{c}} \leq t \wedge \forall j, k.\left(\hat{c} \notin c_j \vee \tau_k^j \notin \left[t_i^{\hat{c}}, t_{i+1}^{\hat{c}}\right)\right)\right\}\right| ,$$

to denote the number of rehearsal requirements that are not naturally satisfied by the visitation schedule during the time interval $[0, t]$.

▶ **Example.** Consider the human computable function $f_{2,2}$ from section 3, and suppose that the user has to compute $f_{2,2}(\sigma(C_i))$ to authenticate at account A_j, where $C_i = (x_0, \ldots, x_{13})$. When the user computes $f_{2,2}$ he must rehearse the associations $(x_{10}, \sigma(x_{10}))$, $(x_{11}, \sigma(x_{11}))$, $(x_{12}, \sigma(x_{12}))$, $(x_{13}, \sigma(x_{13}))$ and $(x_i, \sigma(x_i))$ where $i = (\sigma(x_{10}) + \sigma(x_{11}) \mod 10)$. Thus $c_j \supset \{x_i, x_{10}, x_{11}, x_{12}, x_{13}\}$. When user authenticates he naturally rehearses each of these associations in c_j.

If a cue-association pair $(\hat{c}, \hat{a})$ is not rehearsed naturally during the interval $\left[t_i^{\hat{c}}, t_{i+1}^{\hat{c}}\right)$ then the user needs to perform an extra rehearsal to maintain the association. Intuitively, $ER_{t,\hat{c}}$ denotes the total number of extra rehearsals of the cue-association pair $(\hat{c}, \hat{a})$ during the time interval $[0, t]$, and $ER_t = \sum_{\hat{c} \in C} ER_{t,\hat{c}}$ denotes the total number of extra rehearsals during the time interval $[0, t]$ to maintain all of the cue-association pairs. Thus, a smaller value of $E[ER_t]$ indicates that the user needs to do less extra work to rehearse his secret mapping.

Poisson Arrival Process

The visitation schedule for each account A_j is given by a Poisson arrival process with parameter λ_j, where $1/\lambda_j = E\left[\tau_{j+1}^i - \tau_j^i\right]$ denotes the average time between consecutive visits to account A_j.

Table 7 $\mathbb{E}[ER_{365}]$: Extra Rehearsals over the first year to remember $\sigma : \{1, \ldots, n\} \to \mathbb{Z}_{10}$ in our scheme with $f_{2,2}$ or $f_{1,3}$. Compared with Shared Cues schemes SC-0,SC-1 and SC-2[13].

	Our Scheme ($\sigma \in \mathbb{Z}_{10}^n$)			Shared Cues		
User	$n = 100$	$n = 50$	$n = 30$	SC-0	SC-1	SC-2
Very Active	0.396	0.001	≈ 0	≈ 0	3.93	7.54
Typical	2.14	0.039	≈ 0	≈ 0	10.89	19.89
Occasional	2.50	0.053	≈ 0	≈ 0	22.07	34.23
Infrequent	70.7	22.3	6.1	≈ 2.44	119.77	173.92

Table 8 Single-Digit Challenge Layout. Given a random mapping σ from letters to digits the user can compute $f_{2,2}(\sigma(`C'))$ by executing the following steps (1) Recall $\sigma(`A')$ – the number associated with the letter A, (2) Recall $\sigma(`B')$, (3) Compute $i = \sigma(`A') + \sigma(`B') \mod 10$ – without loss of generality suppose that $i = 8$, (4) Find the letter at index $i$ – 'M' if $i = 8$, (5) Recall $\sigma(`M')$ (6) Recall $\sigma(`C')$ (7) Compute $j = \sigma(`M') + \sigma(`C') \mod 10$ (8) Recall $\sigma(`D')$ (9) Return $j + \sigma(`D') \mod 10$.

A	B	C	D
0	E	5	J
1	F	6	K
2	G	7	L
3	H	8	M
4	I	9	N

Evaluating Usability

Blocki et al. [13] prove the following theorem. Given a sufficient rehearsal schedule and a visitation schedule, Theorem 33 predicts the value of ER_t, the total number of extra rehearsals that a user will need to do to remember all of the cue-association pairs required to reconstruct all of his passwords.

▶ **Theorem 33.** *[13] Let* $i_{\hat{c}}* = \left(\arg\max_x t_x^{\hat{c}} < t\right) - 1$ *then*

$$E[ER_t] = \sum_{\hat{c} \in C} \sum_{i=0}^{i_{\hat{c}}*} \exp\left(- \left(\sum_{\substack{j \ s.t. \\ \hat{c} \in c_j}} \lambda_j \right) \left(t_{i+1}^{\hat{c}} - t_i^{\hat{c}} \right) \right)$$

We use the formula from Theorem 33 to obtain the usability results in Table 7. To evaluate this formula we need to be given the rehearsal requirements, a visitation schedule (λ_i) for each account A_i and a set of public challenges $\vec{C}_i \in (X_{14})^{10}$ for each account A_i. The rehearsal requirements are given by the Expanding Rehearsal Assumption [13] (we use the same association strength parameter $\mathbf{s} = 1$ as Blocki et al. [13]), and the visitation schedules for each user are given in Table 6. We assume that each password is 10 digits long and that the challenges $\vec{C}_i \in (X_{14})^{10}$ are chosen at random by Algorithm 2. Notice that each time the user responds to a single digit challenge he rehearses the secret mapping at five locations (see discussion in Section 3.1). Because the value of $\mathbb{E}[ER_{365}]$ depends on the particular password challenges that we generated for each account, we ran Algorithm 2 and computed the resulting value $\mathbb{E}[ER_{365}]$ one-hundred times. The values in Table 7 represent the mean value of $\mathbb{E}[ER_{365}]$ across all hundred instances.

I Sum of k-Mins

In the basic Hopper-Blum [34] Human Identification Protocol the user memorizes a subset $S \subseteq [n]$ of $k = |S|$ secret indices. A single digit challenge consisted of a vector $x \in \mathbb{Z}_{10}^n$ of n digits and the user responded by with the $\mod 10$ sum of the digits at $k \leq n$ secret locations plus an error term e

$$\sum_{i \in S} x_i \mod 10 + e \ .$$

Typically, the user will set $e = 0$, but occasionally the user is supposed to respond with a completely random digit instead of the correct response (e.g., so that the adversary cannot simply use Gaussian Elimination to find S). Thus, the human user must occasionally generate random numbers in his head to execute the Hopper-Blum protocol. This is potentially problematic because humans are not good at consciously generating random numbers [53, 27, 42]. In fact, hard learning problems like noisy parity might even be easy to learn when humans are providing the source of noise.

Hopper and Blum [34] also proposed a deterministic human identification protocol call sum of k-mins. In this protocol the user memorized k secret pairs (i, j) of indices $S \subseteq [n]^2$. As before a single digit challenge consists of a vector $x \in \mathbb{Z}_{10}^n$ of n digits. However, now the response to the challenge is deterministic

$$\sum_{(i,j) \in S} \min\{x_i x_j\} \mod 10 \ .$$

We observe that for any constant k the protocol is not secure against polynomial time attackers who have seen $O\big(k \cdot \log n\big)$ examples. The adversary can simply enumerate all possible sets S of k pairs and cross out the ones that are inconsistent with the challenge-response pairs he has already seen. Even for larger k (e.g., greater human work) Hopper and Blum [34] observed that the protocol was not secure against an adversary who has seen $O(n^2)$ examples. To see this observe that we can create an indicator variable $y_{i,j}$ for each pair (i, j). Each challenge response pair (x, r) yields a linear constraint

$$\sum_{(i,j)} y_{i,j} \min\{x_i, x_j\} = r \mod 10 \ .$$

Towards Hardness of Approximation for Polynomial Time Problems*

Amir Abboud[1] **and Arturs Backurs**[2]

1 **Stanford University, Palo Alto, USA**
abboud@cs.stanford.edu
2 **MIT, Cambridge, USA**
backurs@mit.edu

Abstract

Proving hardness of approximation is a major challenge in the field of fine-grained complexity and conditional lower bounds in P. How well can the Longest Common Subsequence (LCS) or the Edit Distance be approximated by an algorithm that runs in near-linear time? In this paper, we make progress towards answering these questions. We introduce a framework that exhibits barriers for truly subquadratic and *deterministic* algorithms with good approximation guarantees. Our framework highlights a novel connection between deterministic *approximation algorithms* for natural problems in P and *circuit lower bounds*.

In particular, we discover a curious connection of the following form: if there exists a $\delta > 0$ such that for all $\varepsilon > 0$ there is a deterministic $(1+\varepsilon)$-approximation algorithm for LCS on two sequences of length n over an alphabet of size $n^{o(1)}$ that runs in $O(n^{2-\delta})$ time, then a certain plausible hypothesis is refuted, and the class E^{NP} does not have non-uniform linear size Valiant Series-Parallel circuits. Thus, designing a "truly subquadratic PTAS" for LCS is as hard as resolving an old open question in complexity theory.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases LCS, Edit Distance, Hardness in P

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.11

1 Introduction

Canonical examples of problems that are in P due to natural dynamic programming solutions are Edit Distance and Longest Common Subsequence (LCS) [44, 54]. Given two strings x, y of length n, the LCS problem asks for the length of the longest sequence that appears in both x and y as a (not necessarily contiguous) subsequence, and Edit Distance asks to compute the minimum number of operations (insertion, deletion, or substitution) that is required to transform x into y. Despite decades of attempts, it is not known how to speed up the dynamic programming solution beyond the $O(n^2/\log^2 n)$ bound of Masek and Paterson [66] via the "four Russians" technique. Recent research on the exact complexity of polynomial time problems proved that faster algorithms do not exist [19, 2, 32], not even by polylog factors [5], unless SAT can be solved faster than brute force. A natural question arises: How well can we approximate LCS and Edit Distance in truly subquadratic (or near linear) time? And more generally, can we speed up dynamic programming algorithms without paying too much in the optimality of the solution?

* This work was supported in part by an IBM PhD Fellowship, the NSF and the Simons Foundation.

© Amir Abboud and Arturs Backurs;
licensed under Creative Commons License CC-BY

Various generalizations of LCS and Edit Distance, like the Local Alignment problem [75, 8], are fundamental in computational biology and genomics. In such applications, the input size is a few billions, quadratic time algorithms are prohibitive and are rarely run in practice. To analyze the genome, researchers often use various heuristics, like BLAST [13], that run in near linear time but have no optimality guarantees. Despite BLAST's impact, as witnessed by its more than fifty thousand citations, the bioinformatics community is in an everlasting search of "better" algorithms that are able to reveal new phenomena in the massive amounts of biological data that we currently have (see [77, 63, 49]). The theory community ought to provide guidance: is there hope for fast algorithms with strong guarantees? Is there evidence against a $(1+\varepsilon)$-approximation algorithm that runs in near-linear time, even for the more basic LCS and Edit Distance problems?

We say that an algorithm c-approximates the Edit Distance $ED(S,T)$ of two given sequences S,T if it outputs a value x that is $ED(S,T) \leq x \leq c \cdot ED(S,T)$. Since the Edit Distance is at most n, an n-approximation is trivial. A linear time $\sqrt{n}$-approximation follows from the exact algorithm that computes the Edit Distance in time $O(n+d^2)$ where $d = ED(S,T)$ [65]. Subsequently, this approximation factor has been improved to $n^{3/7}$ by Bar-Yossef et al. [21], then to $n^{1/3+o(1)}$ by Batu et al. [22]. Building on the breakthrough embedding of Edit Distance by Ostrovsky and Rabani [69], Andoni and Onak obtained the first near-linear time algorithm with a *subpolynomial* approximation factor of $2^{\tilde{O}(\sqrt{\log n})}$. Most recently, in FOCS'10, Andoni, Krauthgamer, and Onak [15] significantly improved the approximation to polylogarithmic obtaining an algorithm that runs in time $n^{1+\varepsilon}$ and gives $(\log n)^{O(1/\varepsilon)}$ approximation for every fixed $\varepsilon > 0$. There are many works on approximate Edit Distance in various computational models, see e.g. [68, 15, 38] and the references therein.

While LCS and Edit Distance are closely related, they behave quite differently with respect to approximations, and these clever approximation algorithms for Edit Distance are not known to lead to any nontrivial result for LCS. We say that an algorithm c-approximates the LCS of two given sequences S,T if it outputs a value x that is $\frac{LCS(S,T)}{c} \leq x \leq LCS(S,T)$. A cute observation shows that the LCS of binary sequences can be approximated to a factor of 2 in linear time: the longest common subsequence that is all-zero or all-one is at least half from optimal. Note that a 2 approximation for Edit Distance on binary sequences would be a breakthrough. In general, for an alphabet of size $|\Sigma| = s$, it is easy to get an s-approximation for the LCS and it is a longstanding open question to design an $(s-\delta)$-approximation in near-linear time or even strongly subquadratic time for any constant integer $s \geq 2$ and constant $\delta > 0$. Even though many ideas and heuristics for LCS were designed [43, 27, 48, 45] (see also [68, 28] for surveys), none has proven sufficient to compute an $(s-\delta)$-approximation in strongly subquadratic time. A general tool for speeding up dynamic programming algorithms through a controlled relaxation of the optimality constraint could have great impact for algorithm design. Recently, encouraging positive results along these lines were obtained by Saha [72, 73] for problems related to parsing context-free languages. However, we are still far from understanding, more generally, when and how such speedups are possible.

Proving lower bounds for problems in P, under popular conjectures like the Strong Exponential Time Hypothesis (SETH) [57, 35], is a recent and very active line of work [2, 4, 6, 7, 8, 9, 14, 19, 32, 33, 31, 30, 42, 70, 71, 79, 1, 18, 20, 47, 41, 40]. The known results for LCS and Edit Distance [19, 2, 32, 5] do not imply any non-trivial hardness of approximation, i.e. they only rule out (roughly) $(1+1/n)$-approximations in subquadratic time. Achieving strong hardness of approximations results is often highlighted as an important open question for this line of research, and the general sense of the community is that new ideas that deviate significantly from current techniques might be required.

Our Work

In this paper, we make progress towards the important goal of proving inapproximability results for such fundamental problems in P. We introduce a framework that exhibits barriers for subquadratic and *deterministic* algorithms with good approximation guarantees. Admittedly, the "lower bounds" we obtain for problems like LCS are quite weak and are still far from the upper bounds. Still, they are *much* higher than what we knew before: e.g. instead of the trivial $(1 + 1/n)$-approximation hardness, we can show evidence against $(1 + 1/\text{poly} \log n)$ or even $(1 + o(1))$ approximations. Perhaps more interesting than the statements is the framework itself, which highlights a novel connection between deterministic *approximation algorithms* for natural problems in P and *circuit lower bounds*.

We prove a curious connection of the following form: if there is a truly subquadratic deterministic $(1 + \varepsilon)$-approximation algorithm for LCS on two sequences of length n over an alphabet of size $n^{o(1)}$, then the complexity class E^{NP} does not have non-uniform linear size series parallel circuits[1]. This consequence (explained in more detail below) is widely believed to be true. However, proving it unconditionally would be a breakthrough in complexity theory and the study of non-uniform circuit lower bounds. As stated, this is merely a "difficulty" or a "no-pass" result for LCS, not "hardness". It only shows a "circuit lower bounds" barrier for designing a fast $(1 + o(1))$-approximation algorithm for LCS: it is at least as difficult as resolving a longstanding (and considered to be difficult) open question in circuit complexity. However, we prove a stronger result (Theorem 5 below), which we think should be regarded as a "hardness" result as well, giving evidence that such algorithms for LCS might not exist.

We contribute to the growing body of surprising connections between algorithm design and circuit lower bounds (see the recent survey [85]) [62, 56, 83, 86, 61, 50, 37, 16, 59, 60, 64]. A notable tight connection between faster algorithms for Circuit-SAT and circuit lower bounds was shown by Williams [83, 86]: faster-than-trivial Circuit-SAT algorithms for many circuit classes $\mathcal{C}$ imply interesting new lower bounds against that class. For example, via this connection, Jahanjou, Miles, and Viola [60] show that refuting SETH leads to proving the same lower bound against series-parallel circuits stated above. Abboud et al. [5] go a step further and show that slightly faster algorithms for natural and well-studied problems in P (as opposed to Circuit-SAT) are enough to prove lower bounds against large classes like non-uniform NC. A related intriguing connection is between *derandomization* (of algorithms and circuits) and circuit lower bounds [56, 61, 84, 74, 26, 12]. A derandomization algorithm for a circuit class $\mathcal{C}$ is a deterministic algorithm that is able to distinguish, given a circuit from $\mathcal{C}$, whether it is unsatisfiable (zero satisfying assignments) or "very satisfiable" (at least $2^n \cdot (1 - o(1))$ satisfying assignments). Note that a derandomization algorithm can be obtained from an algorithm that approximates the number of satisfying assignments to circuits from $\mathcal{C}$ (known as CAPP - Circuit Acceptance Probability Problem). Combining the framework of Williams [83] with a "Succinct PCP" [67, 26] shows that to prove a lower bound against a class $\mathcal{C}$ it is enough to obtain a nontrivial *derandomization* algorithm for a class $\mathcal{C}'$ (that could be slightly larger than $\mathcal{C}$) [83, 74, 26]. Our work connects circuit lower bounds, via circuit derandomization tasks, to designing approximation algorithms for natural optimization problems in P like LCS, as opposed to CAPP or algebraic problems like polynomial identity testing (Williams [87] recently showed that derandomizing a quadratic time algorithm for a variant of this problem implies interesting circuit lower bounds).

[1] The class E^{NP} or $\mathsf{TIME}[2^{O(n)}]^{\mathsf{NP}}$ is the class of problems solvable in exponential time with access to an NP oracle.

1.1 Our Results

We will now give a more detailed overview of our results, and then in Section 3 we present the technical details of our framework. Section 1.3 will discuss how our approach could lead to further hardness of approximation results for problems in P. The complete proofs are given in the subsequent sections.

The Gap Block Disjointness Hypothesis

Our result for LCS will be based on the presumed difficulty of solving the following Gap Block Disjointness (GBD) problem in subquadratic time.

▶ **Definition 1** (Gap Block Disjointness)**.** Given two lists of Boolean matrices $A, B \subseteq \{0,1\}^{K \times D}$ of size $|A| = |B| = N$, we say that a pair $A_i \in A, B_j \in B$ is a "good pair" if there exists a $k \in [K]$ such that the rows $A_i(k, \cdot)$ and $B_j(k, \cdot)$ are disjoint, i.e.

$$\vee_{k \in [K]}(\wedge_{h \in [D]}(\neg A_i(k,h) \vee \neg B_j(k,h))) = 0.$$

The Gap Block Disjointness problem is to decide whether we are in case 1 or in case 2 (and if we are in neither, the output can be arbitrary):

1. (zero "good" pairs) none of the pairs $A_i \in A, B_j \in B$ are good.

$$\Pr_{i,j \in [N]}\left[\ \vee_{k \in [K]}(\wedge_{h \in [D]}(\neg A_i(k,h) \vee \neg B_j(k,h))) = 0\ \right] = 0$$

2. (many "good" pairs) at least $N^2 \cdot (1 - 1/\log_2^{10} N)$ pairs $A_i \in A, B_j \in B$ are good.

$$\Pr_{i,j \in [N]}\left[\ \vee_{k \in [K]}(\wedge_{h \in [D]}(\neg A_i(k,h) \vee \neg B_j(k,h))) = 1\ \right] \geq (1 - 1/\log_2^{10} N)$$

A trivial algorithm solves this problem in quadratic time, by going over all pairs of matrices, but can we do better? Note that if we ask whether at least one "good pair" exists (without the above gap-promise) then the problem requires $N^{2-o(1)}$ under SETH (which is conjectured to hold even for randomized algorithms), even when $K = 1$ and $D = \Omega(\log N)$, since this is the Orthogonal Vectors problem [82, 10, 39]. We introduce the hypothesis that this *gap* version, with $D = \Omega(\log N)$ and $K = N^{o(1)}$, cannot be solved by a deterministic algorithm in truly subquadratic time.

▶ **Hypothesis 2.** *There is no $\varepsilon > 0$ and $\alpha > 0$ such that for all constant d we can solve the Gap Block Disjointness problem on binary matrices in $n^\alpha \times d \log n$ in deterministic $O(n^{2-\varepsilon})$ time.*

Interestingly, unlike all previous hardness conjectures in the "Hardness in P" research [52, 70, 79, 6, 9, 55, 3, 11], ours does not remain plausible when faced against randomized algorithms. A near-linear time randomized algorithm that samples a few pairs can easily solve this problem, with high probability. But can a *deterministic* algorithm do anything clever enough to solve the problem in truly subquadratic time? Such an algorithm is not known, and in fact, Lemma 3 below suggests that it would be a breakthrough.

Series-parallel circuits [78, 34, 81, 46] (or VSP circuits) are special kind of circuits that can be obtained by combining circuits either in series or in parallel (defined formally in Section 2). In 1977, Valiant introduced these circuits and argued that most known computer programs fit under this restriction. His hope was that understanding these circuits would be easier than the general case. Four decades later, we still do not know how to resolve basic challenges proposed in his paper, like showing an explicit function that does not have *linear*

size series parallel circuits. It is still conceivable that the large class E^{NP} can be computed by such circuits, and proving otherwise would be a major achievement. Our first lemma states that refuting Hypothesis 2 is at least as difficult as showing these results.

▶ **Lemma 3.** *If Hypothesis 2 is false, then the class* E^{NP} *does not have non-uniform linear size* VSP *circuits.*

A reader familiar with previous SETH lower bounds might wonder why we need this GBD problem, as opposed to simply considering the $K = 1$ case, i.e. the gap version of Orthogonal Vectors. Without going into the details, we remark that, as far as we can show, a faster deterministic algorithm for that case would not imply any new circuit lower bound. Intuitively, this is because the $K = 1$ case can only encode CNF formulas, which is an extremely weak computational model, for which the corresponding circuit lower bounds are easy to prove unconditionally.

We stress that this circuit lower bound consequence is only meant to show that the hypothesis is hard to *refute*. As evidence that the hypothesis is *plausible*, we remark that none of the current (e.g. [39, 53]) or conjectured-to-exist derandomization techniques (e.g. if $\mathsf{P} = \mathsf{BPP}$) are enough to refute it. While a common belief is that randomized algorithms cannot outperform deterministic ones by more than a polynomial factor, it is plausible that randomization can give, say, a linear $\Omega(n)$ speedup.

Reduction to Approximate LCS

The simplicity of the GBD problem makes Hypothesis 2 an appealing starting point for proving barriers. Our main technical lemma shows how GBD can be reduced to LCS while creating a multiplicative gap, giving the first nontrivial hardness of approximation result for LCS.

▶ **Lemma 4.** *If for some* $\delta > 0$*, there is a deterministic algorithm that can approximate the LCS of two given sequences of length* n *over an alphabet of size* $n^{o(1)}$ *to within a* $(1 + \varepsilon)$ *factor, for all* $\varepsilon > 0$*, in* $O(n^{2-\delta})$ *time, then Hypothesis 2 is false.*

Together, these two lemmas imply our main theorem:

▶ **Theorem 5.** *If for some* $\delta > 0$ *there is a deterministic* $(1 + o(1))$*-approximation algorithm for LCS on two sequences of length* n *over an alphabet of size* $n^{o(1)}$ *in* $O(n^{2-\delta})$ *time, then Hypothesis 2 is false and the class* E^{NP} *does not have non-uniform linear size* VSP *circuits.*

We remark that our hardness for approximate LCS immediately transfers to nontrivial results for other problems. For example, we get that the RNA Folding problem which is central in computational biology [2, 17, 51, 76, 80] cannot be approximated to within in a $(1 + o(1))$ factor in truly subquadratic time.

A simple application of our framework, gives a weaker lower bound for Edit Distance and LCS on *binary* sequences. In Section 4.1, we show that a deterministic $(1 + 1/\mathrm{poly}\log n)$-approximation for these problems in truly subquadratic time implies that E^{NP} does not have log-depth (NC^1) circuits.

It is likely that more efficient reductions from GBD to LCS (and Edit Distance) can be devised, and if certain gadgets in our proof can be implemented more efficiently, a *tight* conditional lower bound against $(2 - \delta)$-approximations for LCS on binary sequences could follow. Moreover, better gadgetry would be able to boost the consequence from a VSP circuit lower bound through GBD to a stronger circuit lower bound as in [5]. On a different

note, we believe that a further tightening of the connections between nontrivial circuit derandomization and approximation algorithm for extensively studied problems like LCS and Edit Distance could be a promising direction for *proving* new circuit lower bounds. Perhaps, via stronger connections, one would be able to use highly involved algorithms like Andoni et al.'s approximation for Edit Distance [15] to prove new breakthroughs in complexity theory.

1.2 Technical Overview

To motivate our framework, we give a short exposition of the known constructions for hardness of sequence alignment problems, and why they fail to give any nontrivial consequences of approximation algorithms.

For concreteness, consider the reductions from CNF-SAT to LCS used to prove that LCS requires $n^{2-o(1)}$ time under SETH [2, 32]. In these reductions, we take a CNF formula on n variables and m clauses, say $m = O(n)$, and produce two sequence of length $N \cdot \operatorname{poly}\log N$ where $N = 2^{n/2}$, so that the LCS of the two sequences is large iff the formula is satisfiable. Each sequence is composed of $O(N)$ segments of length $O(\log N)$ called *assignment gadgets*, representing all $2^{n/2}$ partial assignments to half of the variables (each half of the variables is represented in the gadgets in one of the two sequences). When two assignment gadgets are "matched" in an LCS alignment, the contribution to the total score is X_{sat} if the two corresponding partial assignments make up a satisfying assignment to our CNF formula and X_{unsat} otherwise, where $X_{sat} > X_{unsat}$. Due to the way these gadgets are composed, the optimal LCS will be achieved by matching roughly $N = 2^{n/2}$ pairs and therefore gaining a score that is at most $N \cdot X_{unsat}$ if the formula is unsatisfiable, and at least $(N-1) \cdot X_{unsat} + X_{sat}$ if it is satisfiable. Now, notice that X_{sat} cannot be more than $O(\log N)$, since it is upper bounded by the length of the assignment gadgets, while X_{unsat} is at least 1, since otherwise the gadgets do not encode enough information. This implies that the multiplicative gap between the two cases is no more than $(1 + 1/N)$.

A natural attempt to increase this gap is by using a PCP theorem on the initial CNF formula before reducing it to LCS. For instance, this approach has been used in many ETH based lower bounds [29] [2]. Even if an ultra efficient PCP theorem existed, where the number of variables remains $(1 + o(1)) \cdot n$ and the gap increases arbitrarily, this approach does not give any interesting hardness for approximate LCS: The PCP will only affect the gap between X_{sat} and X_{unsat}, which only affects the low order terms in the total score: it is always upper bounded by the length of the gadgets $X_{sat}/X_{unsat} = N^{o(1)}$, and so the multiplicative gap remains $(1 + N^{o(1)}/N)$. The problem with this approach is that the dominant factor is *not* how many clauses our best assignment satisfies, but it is how many *assignments* satisfy all clauses. Standard PCP approaches make unsatisfying assignments less satisfying, but they do not affect the *number of satisfying assignments*.

This leads to another natural attempt: can we find a different kind of gap amplification result that reduces a formula f to another formula f' so that f' has *many* satisfying assignments iff f is satisfiable? To get interesting hardness results for LCS, we will need the gap between the two cases to be quite large, e.g. zero satisfying assignments vs. at least $2^n/10$ satisfying assignments. We would like to do this while keeping the number of variables $(1 + o(1)) \cdot n$. Unfortunately, this kind of gap amplification for CNF's and circuits is unlikely as it would lead to a randomized polynomial time algorithm for CNF-SAT ($\mathsf{BPP} = \mathsf{NP}$): perform the reduction then sample $O(n)$ assignments and check if they satisfy f'.

[2] ETH states that 3-SAT cannot be solved in $2^{o(n)}$ time. Obtaining tight lower bounds for problems in P under ETH is a major open question.

While it is easy for a randomized algorithm to distinguish between unsatisfiable formulas (or circuits) and almost-completely satisfiable ones, the main observation at the base of our framework is that this task might not be so easy for *deterministic* algorithms. Given a circuit on n variables that has $2^n - 2^n/n^{10}$ satisfying assignment, how can a deterministic algorithm find one of its satisfying assignments? If the algorithm treats the circuit as a black-box and blindly queries it with assignments until it outputs 1, the runtime will not be $O(2^{(1-\varepsilon)n})$. Otherwise, the algorithm could try to analyze the circuit and understand its satisfiability properties in order to achieve faster deterministic runtime. The central insight in the connections between algorithms and lower bounds [85] is that our ability to design algorithms for analyzing circuits (from a certain class $\mathcal{C}$) is closely related to our ability to show limitations of circuits (lower bounds against the class $\mathcal{C}$). In fact, there are formal and quite tight connections showing that a faster-than-trivial deterministic algorithm for the "circuit-derandomization" problem above, on certain classes of circuits, implies new circuit lower bounds.

Our framework takes this route in order to get evidence for difficulty and hardness of designing approximation algorithms. Depending on the target problem (in P) for which we seek a "lower bound", one might want to start from a different derandomization problem concerning a different class of circuits so that it embeds as efficiently as possible into the problem. In this paper, we instantiate the framework with the class of linear size series parallel circuits and prove that they embed nicely into approximate LCS (via the GBD problem).

We remark that all our results for consequences of *deterministic* algorithms remain valid for (appropriately defined) *co-nondeterministic* algorithms (see [36] for interesting results on this notion). It is difficult to approximate LCS even with nondeterminism.

Derandomization implies Circuit Lower Bounds

The connection between derandomizing circuits and lower bounds originates in the works of Impagliazzo, Kabanets, and Wigderson [56] and has been optimized significantly by the work of Williams [83], Santhanam and Williams [74], and more recently by Ben-Sasson and Viola [26]. These connections rely on "Succinct PCP" theorems [24, 67, 25, 23, 26], and the recent optimized construction of Ben-Sasson and Viola [26] is crucial to our main result.

Our starting point is the following theorem (Theorem 1.4 in [26]), which we will state more formally later in the paper: Let F_n be a set of functions from $\{0,1\}^n$ to $\{0,1\}$ that satisfies some minor requirements (e.g. functions that can be computed by linear size circuits from some class $\mathcal{C}$). If the acceptance probability of a function of the form

- AND of fan-in $n^{O(1)}$
- of OR's of fan-in 3
- of functions from $F_{n+O(\log n)}$

can be distinguished from being $= 1$ or $\leq 1/n^{10}$ in $2^n/n^{\omega(1)}$ deterministic time, then there is a function f in E^{NP} on n variables such that $f \notin F_n$ (and therefore cannot be computed by linear size circuits from $\mathcal{C}$, and E^{NP} is not contained in non-uniform $\mathcal{C}$).

The optimization of the Succinct PCP by Ben-Sasson and Viola makes the overhead in this connection quite small: we only need two additional levels to the original circuit class (the AND and OR), one of which has fan-in 3. When instantiating this theorem with linear size VSP circuits, this minor overhead allows us to still obtain a simple enough class of circuits that allows for an efficient reduction to our GBD problem (and then to approximate LCS).

Next, we show a reduction from the derandomization task above of AND-OR-VSP circuits to the GBD problem. This reduction is obtained via a series of transformations to the circuit, so that we end up with an OR-AND-OR circuit of the following form, for some constants ℓ, c, μ, which embeds nicely into GBD:

- OR of fan-in $n^{O(1)} \cdot 2^{n/\ell} \cdot 2^{\varepsilon n}$,
- of AND of fan-in $f(\varepsilon, k) \cdot n$ where $k = 2^{2^{\mu c \ell}}$,
- of OR of fan-in k of literals.

To get this form, we use the classical depth reduction of Valiant [78] which is especially powerful for VSP circuits, as well as the sparsification lemma for CNF formulas [57, 58]. The details of Valiant's depth reduction theorem were clarified by Calabro [34] and Viola [81] (cf. Cygan et al. [46]). To reduce the derandomization problem of such AND-OR-AND circuits to GBD we follow the split-and-list technique, similarly to the reduction from CNF-SAT to Orthogonal Vectors [82]. Choosing all the parameters carefully, we get Lemma 3.

Reducing to Approximate LCS

The reduction from GBD to approximate LCS has two main components: an *inner construction*, in which we encode each of the matrices separately into (short) "matrix" or "inner" gadgets, and an *outer construction* that combines the inner gadgets into two final (long) sequences. This outer and inner outline is not different from previous reductions to LCS and Edit Distance, except that now will we have to make sure that our constructions preserve multiplicative gaps. Getting a gap in either of these constructions was beyond previous techniques, and our work contributes to both: The gap in the outer construction will follow, for the most part, from our starting point (GBD and the derandomization problems as opposed to SAT). For the inner construction, however, we need to design new gadgetry that could be of interest even beyond our "difficulty via derandomization" framework. We explain the main ideas below.

The inner construction: In the inner construction we map each one of our input matrices $A_i \in A$ into a sequence a_i and each of our $B_j \in B$ into a sequence b_j so that there is a value X_{bad} and a constant $\varepsilon > 0$ for which: $LCS(a_i, b_j) \leq X_{bad}$ if A_i, B_j is not a good pair, while if A_i, B_j is a good pair then the LCS is much larger $LCS(a_i, b_j) \geq (1 + \varepsilon) \cdot X_{bad}$.

Constructions from previous work [2, 32, 5] easily give such "matrix" gadgets, except the gap between the two cases would be too small: X_{bad} vs $X_{bad} \cdot (1 + 1/n^{\alpha})$. Instead, we introduce some new ideas that lead to a $(1 + \varepsilon)$ gap at the cost of less efficiency in terms of the length of the gadgets and the alphabet size.

We will first construct "disjointness" gadgets that encode each row of each of our matrices. For this discussion, fix a pair of matrices $A_i \in A, B_j \in B$. We will first encode each of A_i's rows $A_i(k, \cdot)$ with a sequence $a_{i,k}$ and each of B_j's rows $B_j(k, \cdot)$ with a sequence $b_{j,k}$ (these are the disjointness gadgets). Our goal will be to have $LCS(a_{i,k}, b_{j,k})$ be $X_{intersect}$ if the two rows intersect, and at least $(1+\varepsilon) \cdot X_{intersect}$ if the rows are disjoint. To achieve this, we use a new encoding that is specifically tailored towards LCS on large alphabets. The idea of our encoding is to chop each of our rows of length $d \log n$ into ℓ pieces of length $d \log n / \ell$ each, and then think of each piece as a separate letter from an alphabet of size $2^{d \log n/\ell} = n^{o(1)}$. Then, we let $a_{i,k}$ be the concatenation of ℓ such symbols, corresponding to the ℓ pieces that appear in the row $A_i(k, \cdot)$. Meanwhile, $b_{j,k}$ will be defined differently: for each piece x, we will enumerate all possible letters σ that correspond to pieces y that are disjoint from x, and write them in a dedicated segment in $b_{j,k}$. By doing this, the LCS of $a_{i,k}$ and $b_{j,k}$ will be exactly ℓ if the two rows are disjoint (since all pieces will be disjoint and contribute a letter

to the LCS), while if the rows intersect the LCS will be at most $\ell - 1$. Since we can afford to pick ℓ to be a (large enough) constant, we obtain a multiplication gap of $(1 + \varepsilon)$ where $\varepsilon = 1/(\ell - 1)$ is a constant.

Next, we need to combine these "disjointness" gadgets into "matrix gadgets" with an OR: we want the score to be large iff *there exists* a $k \in [K]$ for which the rows are disjoint. Previously known OR gadgets are not sufficient: the total score would contain a sum over all $k \in [K]$ of the score of the k^{th} disjointness gadget, which would decrease the gap back to $(1 + 1/K) = (1 + 1/n^{\alpha})$. To overcome this, we use a different OR gadget that heavily abuses the alphabet in order to keep the multiplicative gap *unchanged*. The idea is simple: let us designate a separate alphabet Σ_k for the k^{th} disjointness gadget $a_{i,k}$ or $b_{j,k}$ (representing the k^{th} row of A_i or B_j), so that for $k \neq k'$ the alphabets are disjoint $\Sigma_k \cap \Sigma_{k'} = \emptyset$. And now our matrix gadgets, which are an OR of our disjointness gadgets, are defined as follows:

$$a_i := a_{i,1}\ a_{i,2}\ \cdots\ a_{i,k}$$

$$b_j := a_{j,k}\ b_{j,k-1}\ \cdots\ b_{j,1}$$

The extremely useful property of this construction is that the LCS is the *maximum* over k of the LCS of $a_{i,k}$ and $b_{j,k}$, as opposed to any expression with a summation over all k. To see this, first note that only letters from gadgets with the same index k can be matched, and now imagine we pick at least one matching for some k, say for $k = 1$ so that we matched some letter between $a_{i,1}$ and $b_{j,1}$, and notice that now we can no longer match any letters from gadgets with a different index $k' \neq k$ without creating a crossing. Therefore, we get that the score of these matrix gadgets is *exactly the same* as the score of the best disjointness gadget across all rows $k \in [K]$, and so if A_i, B_j is a bad pair the score cannot be more than $X_{bad} = X_{intersect}$, while if the pair is good the score is $X_{good} = X_{disjoint} \geq (1+\varepsilon) \cdot X_{intersect}$, where ε is the same constant defined above.

For the outer construction, we use a similar "alignment gadget" to the one used in previous works [2, 32], but we need to analyze it more carefully in order to argue that it generates a gap when working with a gap problem like GBD. We show that if we are in case 2, then there is a matching that contains a large number of pairs A_i, B_j each contributing $(1 + \varepsilon) \cdot X_{bad}$, while in case 1 all pairs in all optimal matchings will contribute only X_{bad}. While previous work used padding of size that is linear in the size of the inner gadgets, we will have to work with much smaller paddings of size that is linear only in the LCS between the inner gadgets. By a careful choice of the parameters we show that this is a $(1 + \varepsilon')$ gap, for some $\varepsilon' > 0$.

1.3 Discussion

Fundamentally, our approach is based on the following intuition. If there is a search problem that we do not expect any (deterministic or randomized) algorithm to be able to solve much faster than brute force, like the problem of finding a pair of vectors that satisfy some function (as in GBD and Orthogonal Vectors), then we might expect the *gap* version of the problem to be hard for deterministic algorithms: maybe we cannot even distinguish the case in which no "good solutions" exist in the search space from the case that almost any solution is "good".

In the context of circuit lower bound consequences from circuit analysis algorithms, this intuition is more or less formal: most lower bound consequences we can get from SAT algorithm follow also from such a distinguisher.

Does the same hold with respect to the other conjectures used in "Hardness in P" research? Consider the 3-SUM problem which asks if a set of n numbers contains three that sum to zero, and is conjectured to require $n^{2-o(1)}$ time. If we believe the 3-SUM conjecture, should

we also believe that we cannot deterministically distinguish an input with many triples that sum to zero from an instance with few? Would this "Gap 3-SUM Conjecture" have interesting consequences? We believe that this is an intriguing avenue for future research and expect it to be fruitful, either in terms of conditional lower bounds, or in terms of a better understanding of our conjectured-to-be-hard problems.

2 Valiant series-parallel circuits

▶ **Definition 6** (Valiant series-parallel graphs [78, 34, 81, 46]). A *multidag* $G = (V, E)$ is a directed acyclic multigraph. Let input(G) be the set of vertices of G with in-degree 0. Let output(G) be the set of vertices of G with out-degree 0. We say that the multidag G is a Valiant series parallel (VSP) graph if there exists a *labelling* $l : V \to \mathbb{Z}$ of G with the following properties:

- For all directed edges $(u, v) \in E$ we have that $l(u) < l(v)$.
- There exists an integer $d \in \mathbb{Z}$ such that for all $v \in$ input(G), $l(v) = d$. The definition from [34] asks that $d = 0$. It is not hard to verify that our definition is equivalent to theirs.
- There exists an integer $d' \in \mathbb{Z}$ such that for all $v \in$ output(G), $l(v) = d'$.
- There is *no* pair of directed edges $(u, v), (u', v') \in E$ such that the inequality $l(u) < l(u') < l(v) < l(v')$ holds.

▶ **Definition 7** (Valiant series-parallel circuits [78, 34, 81, 46]). A circuit is a Valiant series-parallel circuit if the underlying multidag is a VSP graph and the fan-in of every gate is at most 2.

▶ **Definition 8** (Size of a circuit). The size of a circuit on n input variables is equal to the number of gates in it. We do not count the $n + 2$ input nodes, i.e. the input variables and the two constant values 0 and 1 (which are assumed to be given as the last two input nodes to a circuit).

▶ **Definition 9** (VSP_c). We define class VSP_c to be the set of languages recognizable by VSP circuits of size at most $\leq cn$ where n is the number of input variables. The set of allowed gates is the set of *all* gates of fan-in at most 2.

Below we show properties of the class VSP_c that we will use later in the paper.
We need the following definition from [26].

▶ **Definition 10** ([26]). Let F_n be a set of functions from $\{0,1\}^n$ to $\{0,1\}$. We say that F_n is *efficiently closed under projections* if functions in F_n have a poly(n)-size description and given (the description of) a function $f \in F_n$, indexes $i, j \leq n$, and a bit b, we can compute in time poly(n) the functions $\neg f$, $f(x_1, \ldots, x_{i-1}, b \text{ XOR } x_j, x_{i+1}, \ldots, x_n)$, and $f(x_1, \ldots, x_{i-1}, b, x_{i+1}, \ldots, x_n)$, all of which are in F_n.

▶ **Lemma 11.** *The class* VSP_c *is efficiently closed under projections for any constant* $c > 1$.

Proof. From Definition 9 it follows that the class VSP_c has poly(n) description: the circuit itself. Consider a function f on n input variables from VSP_c that has a VSP circuit of size at most $\leq cn$. We show that the three functions from the statement of Definition 10 can be computed in poly(n) time and that all of them are in VSP_c.

Function $\neg f$

Consider the output. If it is one of the inputs, we add a NOT gate and remove all the other gates. If the output is not one of the inputs, it must be some gate g. We replace it with gate $\neg g$. Since allow *all* gates of fan-in at most 2, we can do this. The number of gates did not increase and the function $\neg f$ is now in VSP_c. Clearly, the transformation can be done in poly(n) time.

Function $f(x_1, \ldots, x_{i-1}, b \text{ XOR } x_j, x_{i+1}, \ldots, x_n)$

If $b = 0$, we rewire all gates that used input x_i to use input x_j. If $b = 1$, we rewire all gates to use NOT x_j. Since we have all gates of fan in at most 2, we don't need to introduce the NOT gate. Instead, we replace the gate by another gate that negates the corresponding input. Similarly as before, the transformation can be done in poly(n) time and we did not increase the number of gates. Thus, the resulting function is in VSP_c.

Function $f(x_1, \ldots, x_{i-1}, b, x_{i+1}, \ldots, x_n)$

The transformation is similar as in the previous case. Instead of rewiring to x_j, we rewire to the constant function 0 which is among the inputs. ◀

▶ **Lemma 12.** *Let $f_1, f_2, f_3 \in \mathsf{VSP}_c$ be three functions on n variables from the class VSP_c. Then the function*

$$f := \neg(f_1 \ OR \ f_2 \ OR \ f_3)$$

on the same n variables belongs to VSP_{4c} if $c \geq 4$ and $n \geq 10$.

Proof. For every $i = 1, 2, 3$, let C_i be the VSP circuit of size $\leq cn$ corresponding to the function f_i, and let G_i be the underlying VSP multidag of C_i. Let the multidag $G := P(G_1, G_2, G_3)$ be the *disjoint* union of the underlying VSP multidags G_1, G_2, G_3, and let $\text{input}(G_i) = \{u_i^1, \ldots, u_i^n, u_i^{n+1}, u_i^{n+2}\}$ be the $n + 2$ input nodes for C_i, $i = 1, 2, 3$ (see Definition 8), where the first n nodes $u_i^1, \ldots, u_i^n$ correspond to the n input variables, and u_i^{n+1} and u_i^{n+2} correspond to the two constant functions 0 and 1, respectively. We have that $\text{input}(G) = \text{input}(G_1) \cup \text{input}(G_2) \cup \text{input}(G_3)$. Moreover, since $|\text{input}(G_i)| = n + 2$, $|\text{input}(G)| = 3n + 6$. Each C_i has only one output gate. Thus, $\text{output}(G_i) = \{o_i\}$ for some node o_i. Therefore, $|\text{output}(G)| = 3$.

A disjoint union of two VSP multidags is a multidag (see the proof of Lemma 3 in [34]). Therefore, the multidag G is a VSP multidag. Let, then, l be the labeling of G according to Definition 6 of VSP graphs. We construct a circuit C for the function f as follows. First, we let C be the disjoint union of C_1, C_2, C_3 (each C_i has its own $n + 2$ input nodes). Therefore, the underlying graph of C is G. Next, whenever we add a node or an edge to G, we do the same for C, and the other way around. As the circuits C_1, C_2, C_3 do not share their inputs, we add $n + 2$ input nodes $u^1, \ldots, u^n, u^{n+1}, u^{n+2}$ to C (and to G). The first n input nodes $u^1, \ldots, u^n$ correspond to the n input variables, and the 2 input nodes u^{n+1} and u^{n+2} correspond to the two constant function 0 and 1, respectively. We connect the n input nodes $u^1, \ldots, u^n$ in pairs to the first n input nodes of C_i (for every $i = 1, 2, 3$). That is, for every $j = 1, \ldots, n$ and $i = 1, 2, 3$, we connect u^j to u_i^j. In addition, for every $j = n + 1, n + 2$ and $i = 1, 2, 3$, we connect u^j to u_i^j.

For every newly added input node u^j, $j = 1, \ldots, n+2$, we update the labeling: $l(u^j) = d-1$. As a result, the multidag G has $\text{input}(G) = \{u^1, \ldots, u^n, u^{n+1}, u^{n+2}\}$ and all weights of these

nodes are equal to $d-1$. It remain to verify the fourth property of VSP graphs. Since for every u^j, if (u^j, v) is an edge in G, then $l(v) = d$, the fourth property also holds. Thus G is VSP graph.

Now we have three functions f_1, f_2, f_3 on the same set of $n+2$ inputs. To get the function $f = \neg(f_1 \text{ OR } f_2 \text{ OR } f_3)$, we add two more gates u_1, u_2 to the circuit C. We set the labeling: $l(u_1) = d' + 1$ and $l(u_2) = d' + 2$. u_1 is an OR gate, and it computes the OR of o_1 and o_2 (the outputs of the functions f_1 and f_2). The gate u_2 is a $\neg$OR gate, and it computes the negation of the OR of u_1 (the OR of f_1 and f_2) and o_3 (the output of the function f_3). Since all gates of fan-in at most two are allowed, we can implement a $\neg$OR gate. We can check that C computes f (the negation of the OR of f_1, f_2, f_3). The size of the circuit C is at most $3cn + |\text{input}(G)| + 2 = 3cn + 3n + 8 \leq 4cn$ as required. It is not hard to verify that the resulting labeling of u_1, u_2 and the rest of the multigraph G satisfies the properties from Definition 6. Thus, we conclude that the resulting underlying multidag G is a VSP graph and that C is a VSP_{4cn} circuit. ◀

3 VSP Circuits and Block Disjointness

Circuit Lower Bounds from Derandomization

The connection between derandomizing circuits and lower bounds originates in the works of Impagliazzo, Kabanets, and Wigderson [56] and has been optimized significantly by the work of Williams [83], Santhanam and Williams [74], and more recently by Ben-Sasson and Viola [26]. These connections rely on "Succinct PCP" theorems [67, 26], and the recent optimized construction of Ben-Sasson and Viola [26] is crucial to our main result. Our starting point is the following theorem.

▶ **Theorem 13** (Theorem 1.4 in [26]). *Let F_n be a set of function from $\{0,1\}^n$ to $\{0,1\}$ that are efficiently closed under projections (see Definition 10).*

If the acceptance probability of a function of the form

- *AND of fan-in $n^{O(1)}$ of*
- *OR's of fan-in 3 of*
- *functions from $F_{n+O(\log n)}$*

can be distinguished from being $= 1$ or $\leq 1/n^{10}$ in TIME$(2^n/n^{\omega(1)})$, then there is a function f in E^{NP} on n variables such that $f \notin F_n$.

The optimization of the Succinct PCP by Ben-Sasson and Viola makes the overhead in this connection quite small: only two additional levels to the circuit, one of which has fan-in 3. Next, we instantiate this theorem with VSP circuits and then do simple tricks to the circuits in order to simplify the derandomization task as much as possible.

▶ **Lemma 14.** *To prove that E^{NP} does not have non-uniform Valiant series parallel* (VSP) *circuits of size cn on n input variables, it is enough to show a deterministic algorithm for the following Derand-VSP problem that runs in $2^n/n^{\omega(1)}$ time. Given a circuit over n input variables of the form:*

- *OR of fan-in $n^{O(1)}$ of*
- *negations of OR's of fan-in 3 of*
- VSP *circuits of size cn,*

distinguish between the case where no assignments satisfy it, versus the case in which at least a $\geq 1 - 1/n^{10}$ fraction of the assignments satisfy it.

Lemma 14 follows from Theorem 13 almost directly: By Lemma 11, the class VSP_c (of functions recognizable by VSP circuits of size $\leq cn$) is efficiently closed under projections. Therefore, we can instantiate Theorem 13 on VSP_c. Since distinguishing the acceptance probability from being $= 1$ or $\leq 1/n^{10}$ is equivalent to distinguishing the *rejection* probability from being $= 0$ or $\geq 1 - 1/n^{10}$, we get Lemma 14 by *negating* the function which is AND of OR of $F_{n+O(\log n)}$ and using De Morgan's law on the AND. W.l.o.g. we replace the number of inputs $n + O(\log n)$ by n.

From Derandomizing VSP Circuits to Gap Block Disjointness

Let C be the circuit on n variables given as an input to the Derand-VSP problem described in Lemma 14. We use known results in complexity theory to convert this circuit into a simpler form that will be easier to work with when reducing to other problems. By Lemma 12, the circuit C can be interpreted as:

- OR of fan-in $n^{O(1)}$ of
- VSP circuits of size $\leq 4cn$,

where the $n^{O(1)}$ VSP circuits use the same set of n inputs.

Next, we use the following classical theorem of Valiant to convert each of these VSP circuits into an OR of CNF's on our n inputs. The ideas in the proof are due to Valiant [78], but the details were shown by Calabro [34] and Viola [81] (cf. Cygan et al. [46]).

▶ **Theorem 15** (Depth reduction [78]). *For all $\ell \geq 1$, we can convert any VSP of size $4cn$ on n variables into an equivalent formula which is OR of $2^{n/\ell}$ k-CNF's on the same n variables, where $k = 2^{2^{\mu c \ell}}$ for some absolute constant $\mu > 0$. The reduction runs in $2^{n/\ell} \cdot n^{O(1)}$ time for any constants c and l.*

▶ Remark. Let $\varepsilon > 0$ an arbitrary constant. Given a circuit on n variables with fan-in 2 gates, of size $O(n)$ and $O(\log n)$ depth, we can transform it into an equivalent formula which is *OR* of $2^{O(\frac{\log n}{\log \log n})}$ CNFs with clause size $\leq n^{\epsilon}$ [78]. However, we can't use this result for our purposes because it will be crucial for us that the clause size in the statement of Theorem 15 is upper bounded by a *constant*.

We will also need to apply the sparsification lemma [57, 58].

▶ **Lemma 16.** *For all $k \geq 3$ and $\varepsilon > 0$ we can convert a k-CNF formula on n variables into an equivalent OR of $2^{\varepsilon n}$ k-CNF formulas on the same variables where each CNF has $f(\varepsilon, k) \cdot n$ clauses, where $f(\varepsilon, k) = (k/\varepsilon)^{O(k)}$.*

Combining all these transformations allows us to focus on circuits of the following OR-AND-OR form. By the following claim, to solve the Derand-VSP problem it is enough to distinguish between the case in which no assignments satisfy a formula of the above OR-AND-OR form and the case in which at least $2^n - 2^n/n^{10}$ assignments do satisfy it.

▶ **Claim 17.** *Let C be an input circuit to the Derand-VSP problem (as described in Lemma 14). For all $\ell \geq 1$ and $\varepsilon > 0$, we can convert C into an* equivalent *formula C' on the same set of n inputs of the following form:*

- *OR of fan-in $n^{O(1)} \cdot 2^{n/\ell} \cdot 2^{\varepsilon n}$, of*
- *AND of fan-in $f(\varepsilon, k) \cdot n$ where $k = 2^{2^{\mu c \ell}}$, of*
- *OR of fan-in k of literals.*

Proof. Recall that an input circuit to the Derand-VSP problem has the form of an OR of fan-in $n^{O(1)}$ of series parallel circuits of size $\leq 4cn$. We want to decide if C is unsatisfiable or at least a $1 - 1/n^{10}$ fraction of the assignments satisfy it. First, we apply Theorem 15 on

every VSP circuit of size $\leq 4cn$. This produces a formula which is an OR of $2^{n/l}$ k-CNFs. Then, we apply the sparsification of Lemma 16 on every k-CNF to obtain a circuit as in the statement of the claim. ◀

This OR-AND-OR form motivates the definition of our Gap Block Disjointness problem (see Definition 1 in Section 1.1). Recall that our GBD Hypothesis (Hypothesis 2 in Section 1.1) states that GBD cannot be solved in truly subquadratic time with a deterministic algorithm. We are now ready to prove that refuting our hypothesis implies a circuit lower bound against linear size VSP circuits, thus establishing a "circuit lower bounds barrier" for refuting our hypothesis. The following claim implies Lemma 3 from Section 1.1.

▶ **Claim 18.** *For all $c \geq 1$ and $\alpha > 0$, there exists a constant $d \geq 1$ such that if there is a deterministic algorithm that solves the Gap Block Disjointness problem on two lists of size N of binary $N^\alpha \times d \log N$ matrices in $N^2/\log^{\omega(1)} N$ time, then E^{NP} does not have non-uniform VSP circuits of size cm (m is the number of input variables). The constant d can be upper bounded by*

$$d \leq 2^{2^{2^{O(c/\alpha)}}}.$$

Proof. By Theorem 13, to show that E^{NP} does not have non-uniform VSP circuits of size cm, it suffices to solve the Derand-VSP problem on a circuit C with $n = m + O(\log m)$ variables in time $2^n/n^{\omega(1)}$.

First, by Claim 17, we can transform the circuit C into an equivalent formula C' of form OR-AND-OR (as described in the statement). Then, we show a reduction from the Derand-VSP problem on the formula C' to the Block Disjoint Pairs problem with the required parameters, as follows. Let $N := 2^{n/2}$. We apply the transformation from Claim 17 to C, with parameters $\varepsilon := \frac{\alpha}{6}$, $l := \frac{6}{\alpha}$, and $d := 2f(\varepsilon, k) \leq (k/\varepsilon)^{O(k)}$, and get an equivalent formula C' of the following form:

- OR of fan-in $n^{O(1)} \cdot 2^{n/\ell} \cdot 2^{\varepsilon n} \leq 2^{\alpha n/2} = N^\alpha$, of
- AND of fan-in $f(\varepsilon, k) \cdot n = d \cdot \log N \leq (k/\alpha)^{O(k)} \cdot n$ where $k = 2^{2^{\mu c \ell}} \leq 2^{2^{O(c/\alpha)}}$, of
- OR of fan-in k of literals.

We think of the formula C' as a disjunction of CNF's with clause size k.

Let us now transform C' to an instance of the Block Disjoint Pairs problem. C' has n binary input variables $x_1, \ldots, x_n$. We split these variables into two parts: $x_1, \ldots, x_{n/2}$ and $x_{1+(n/2)}, \ldots, x_n$, and construct two sets A and B of matrices for the Block Disjoint Pairs problem.

Set of matrices A

Consider all the $N = 2^{n/2}$ partial assignments of the first half $x_1, \ldots, x_{n/2}$ of the variables. We will construct a matrix A_i, $i = 1, \ldots N$, for each partial assignment p_i of $x_1, \ldots, x_{n/2}$ as follows. For every k-CNF in C' we have a corresponding row in A_i, such that every clause of the k-CNF has a corresponding column. Thus, for $r = 1, \ldots, N^\alpha$, the r-th row of the matrix A_i corresponds to the r-th k-CNF in C', and every clause of the r-th k-CNF has a corresponding column in the r-th row such that the t-th clause corresponds to the t-th column in the r-th row of A_i. We set $A_i[r, t]$ to 0 if p_i satisfies the t-th clause of the r-th k-CNF, and to 1 otherwise. A clause is satisfied by a *partial* assignment iff it is satisfied independently of the assignment of the rest of the variables. We assume that the number of k-CNFs is N^α and the number of clauses in each k-CNF is $d \cdot \log N$. If this is not the case, then we can add dummy k-CNFs that are not satisfiable, or clauses that are satisfied by any partial assignment.

Set of matrices B

The second set of matrices B is constructed like the set A but with the second half of variables $x_{1+(n/2)}, \ldots, x_n$.

Our construction satisfies all the parameters of the Block Disjoint Pairs problem. In particular, $d \leq (k/\varepsilon)^{O(k)} \leq 2^{2^{2^{O(c/\alpha)}}}$.

Correctness of the reduction

To prove the correctness of our reduction, it suffices to show that the fraction of pairs of matrices that form a satisfying assignment (the first condition in Definition 1), is the same as the fraction of assignments that satisfy the circuit C'. We show that the i-th partial assignment of $x_1, \ldots, x_{n/2}$ and the j-th partial assignment of $x_{1+(n/2)}, \ldots, x_n$ satisfy C' iff the matrices A_i and B_j form a satisfying assignment too. If C' is satisfied, then at least one of the k-CNFs in C' is satisfied. Assume, then, without loss of generality, that the r-th k-CNF is satisfied. Our goal is to show that $\wedge_{h \in [d \log N]}(\neg A_i(r,h) \vee \neg B_j(r,h)) = True$. Or, equivalently, that $A_i(r,h) \cdot B_j(r,h) = 0$, for every $h \in [d \log N]$. In fact, this follows from the fact that the r-th k-CNF is satisfied and from the construction of A_i and B_j. If, on the other hand, $\vee_{k \in [N^\alpha]}(\wedge_{h \in [d \log N]}(\neg A_i(k,h) \vee \neg B_j(k,h))) = False$, then the i-th partial assignment of $x_1, \ldots, x_{n/2}$ and the j-th partial assignment of $x_{1+(n/2)}, \ldots, x_n$ do not satisfy C'. This follows from the construction of A_i and B_j and the fact that *no* k-CNF is satisfied in this case.

Therefore, $1 - 1/\log_2^{10} N = 1 - 2^{10}/n^{10} \leq 1 - 1/n^{10}$, concluding the proof. ◀

4 The Reduction to Approximate LCS

Our main technical contribution is a reduction from Gap Block Disjointness to $(1+\varepsilon)$ approximate LCS. Lemma 4 from Section 1.1 follows from the following claim, and the rest of this section is dedicated to its proof. For a high level intuition of the reduction see the introduction.

▶ **Claim 19** (Main). *If for some $\delta > 0$, there is a deterministic algorithm that can approximate the LCS of two given sequences of length n over an alphabet of size $n^{o(1)}$ to within a $(1+\varepsilon)$ factor, for all $\varepsilon > 0$, in $O(n^{2-\delta})$ time, then Hypothesis 2 is false.*

Weighted LCS

A natural generalization of LCS that will be useful in our proof is the *weighted longest common subsequence* (WLCS), where each symbol s has a positive integer weight $w(s)$. The weight of a subsequence is the total weight of the symbols it contains, and the WLCS score is the maximum total weight that we can obtain if we range over all common subsequences. As long as the weights are not too large, WLCS and LCS are computationally equivalent due the following lemma.

▶ **Lemma 20** (Lemma 2 in [2]). *Given two weighted sequences x and y, $WLCS(x,y) = LCS(x',y')$, where $|x'| = \sum_{i=1}^{|x|} w(x_i)$ and $|y'| = \sum_{i=1}^{|y|} w(y_i)$. The construction time of x', y' is $O(\max(|x'|,|y'|))$. $|z|$ denotes the length of the sequence z.*

Below, we will use the terms LCS and WLCS interchangeably, and if we do not specify the weight of a symbol s, then it is assumed that $w(s) = 1$.

The parameters of the reduction

We will show the following statement which implies what we need.

If we have a deterministic algorithm for the LCS problem that runs in $O(n^{2-\varepsilon})$ time and that gives $(1+(\delta/10^5))$ approximation, then we can solve the Block Disjoint Pairs problem in $O(N^{2-(\varepsilon/2)})$ time on binary matrices of size $N^\alpha \times d \log N$ for $\alpha := \varepsilon/10$ and $d := \alpha(1+\delta)/\delta$. Choosing $\delta = o(1)$ implies $d = \omega(1)$ and proves the theorem. W.l.o.g. we assume that $\delta \geq 1/\log n$ and $\varepsilon \leq 1/100$.

We will show the statement by providing a deterministic reduction from the Block Disjoint Pairs problem to the LCS problem. We will take the first set of N matrices $A = \{A_1, \ldots, A_N\}$ (each matrix is of size $N^\alpha \times d \log N$) and produce a sequence x of symbols. The sequence x is of length $|x| \leq O(N^{1+2\alpha}d/\alpha) =: n$. Similarly, we will take the second set of N matrices $B = \{B_1, \ldots, B_N\}$ and produce a sequence y of symbols, $|y| \leq O(N^{1+2\alpha}d/\alpha) = n$. The sequences x and y have the property that $LCS(x,y) \leq T$ if we are in Case 1 of Block Disjoint Pairs problem on sets of matrices A and B (see Definition 1) and $LCS(x,y) \geq (1+(\delta/10^5))T$ if we are in Case 2. T is some fixed value. We run the deterministic approximation algorithm for the LCS problem and decide in which case we are. Since the reduction is deterministic, this gives a deterministic algorithm for the Block Disjoint Pairs problem that runs in time

$$O(n^{2-\varepsilon}) \leq O\left(\left(N^{1+2\alpha}d/\alpha\right)^{2-\varepsilon}\right) \leq O\left(\left(N^{1+(\varepsilon/5)}/\delta\right)^{2-\varepsilon}\right),$$

where we use the fact that $\alpha = \varepsilon/10$ and $d = \alpha(1+\delta)/\delta$. Since $\delta \geq 1/\log n$ and $\varepsilon \leq 1/100$, we get that the runtime is upper bounded by $O(N^{(1+(\varepsilon/5))(2-\varepsilon)} \log^2 N) \leq O\left(N^{2-(\varepsilon/2)}\right)$ as required.

Construction of *inner gadgets*

To construct sequences x and y, we need *inner gadgets* $IG(A_i)$, $IG(B_j)$ for every set $A_i \in A$ and $B_j \in B$. We want that $IG(A_i)$ and $IG(B_j)$ satisfy the property that $LCS(IG(A_i), IG(B_j)) = T'$ if the pair A_i, B_j is not satisfying and $LCS(IG(A_i), IG(B_j)) = (1+\delta)T'$ if the pair A_i, B_j is satisfying. Below we will construct such inner gadgets with $|IG(A_i)|, |IG(B_j)| \leq O(N^{2\alpha}d/\alpha)$. After that we will construct the final sequences x and y with the required properties by putting together inner gadgets for all matrices in A and B.

We construct the inner gadgets $IG(A_i)$, $IG(B_j)$ by constructing *disjointness gadgets* $DG(A_i(k,\cdot))$, $DG(B_j(k,\cdot))$ for every row $k \in [K]$ of matrices A_i and B_j.

Properties of the disjointness gadgets

The disjointness gadgets will satisfy the following properties

- For every $k \in [K]$, $|DG(A_i(k,\cdot))|, |DG(B_j(k,\cdot))| \leq O(N^\alpha d/\alpha)$.
- If $\wedge_{h\in[D]}(\neg A_i(k,h) \vee \neg B_j(k,h)) = False$, then $LCS(DG(A_i(k,\cdot)), DG(B_j(k,\cdot))) = T'$. Otherwise, $LCS(DG(A_i(k,\cdot)), DG(B_j(k,\cdot))) = (1+\delta)T'$.
- For every $k \in [K]$, $DG(A_i(k,\cdot)), DG(B_j(k,\cdot)) \in \Sigma_k^*$. $\Sigma_k \cap \Sigma_{k'} = \emptyset$ for $k \neq k'$.

Given such disjointness gadgets, we construct inner gadgets $IG(A_i)$, $IG(B_j)$ as follows:

$$IG(A_i) := \bigcirc_{k=1}^{K} DG(A_i(k,\cdot)) = A_i(1,\cdot) \circ A_i(2,\cdot) \circ A_i(3,\cdot) \circ \ldots \circ A_i(k,\cdot),$$

$$IG(B_j) := \bigcirc_{k=1}^{K} DG(B_j(K+1-k,\cdot)) = B_j(k,\cdot) \circ B_j(k-1,\cdot) \circ B_j(k-2,\cdot) \circ \ldots \circ B_j(1,\cdot).$$

Since $\Sigma_k \cap \Sigma_{k'} = \emptyset$ for $k \neq k'$, the only way to get $LCS(IG(A_i), IG(B_j)) > 0$ is by matching symbols in $A_i(k,\cdot)$ and $B_j(k,\cdot)$. By the construction of $IG(A_i)$ and $IG(B_j)$, if we match

symbols between $A_i(k,\cdot)$ and $B_j(k,\cdot)$, then we can't match symbols between $A_i(k',\cdot)$ and $B_j(k',\cdot)$ if $k' \neq k$. This means that

$$LCS(IG(A_i), IG(B_j)) = \max_{k=1}^{K} LCS(DG(A_i(k,\cdot)), DG(B_j(k,\cdot))).$$

From the properties of disjointness gadgets, we get the required properties of the inner gadgets.

Construction of the disjointness gadgets

Now we will construct the disjointness gadgets $DG(A_i(k,\cdot)), DG(B_j(k,\cdot))$. $A_i(k,\cdot)$ is a binary vector of length $d \log N$. We split it into d/α binary vectors $v_t \in \{0,1\}^{\alpha \log N}$ each of length $|v_t| = \alpha \log N$: $A_i(k,\cdot) = v_1 \dots v_{d/\alpha}$. We define

$$DG(A_i(k,\cdot)) := c_k \circ \bigcirc_{t=1}^{d/\alpha} s_{k,t,v_t},$$

where we set $w(c_k) := (d/\alpha) - 1$. s_{k,t,v_t} are symbols indexed by rows k, indices of vectors t and vectors v_t. We have that

$$DG(A_i(k,\cdot)) \in \Sigma_k^* := \{c_k\} \cup \{s_{k,t,v} \mid v \in \{0,1\}^{\alpha \log N} \text{ and } t \in [d/\alpha]\}.$$

Similarly we split the binary vector $B_j(k,\cdot)$ of length $d \log N$ into d/α binary vectors $w_t \in \{0,1\}^{\alpha \log N}$ each of length $|w_t| = \alpha \log N$: $B_j(k,\cdot) = w_1 \dots w_{d/\alpha}$. We define

$$DG(B_j(k,\cdot)) := \left(\bigcirc_{t=1}^{d/\alpha} \bigcirc_{v \,:\, v \cdot w_t = 0} s_{k,t,v} \right) \circ c_k,$$

where we do the inner product $v \cdot w_t$ over the integers (not modulo 2). Notice that $|DG(A_i(k,\cdot))| \leq O(d/\alpha)$ and $|DG(B_j(k,\cdot))| \leq O(N^\alpha d/\alpha)$ as required.

We claim that if $\wedge_{h \in [D]}(\neg A_i(k,h) \vee \neg B_j(k,h)) = False$, then

$$LCS(DG(A_i(k,\cdot)), DG(B_j(k,\cdot))) = (d/\alpha) - 1$$

and $LCS(DG(A_i(k,\cdot)), DG(B_j(k,\cdot))) = d/\alpha$ otherwise. Since $d = \alpha(1+\delta)/\delta$, we have that $T' = (d/\alpha) - 1$ satisfies the second property of the disjointness gadgets. We now show the claim.

Clearly, $LCS(DG(A_i(k,\cdot)), DG(B_j(k,\cdot))) \geq (d/\alpha) - 1$ because we can match the symbols c_k. Also, we have the equality if we match the symbols c_k in the optimal alignment. Suppose that we don't match c_k. Then it's not hard to check that $LCS(DG(A_i(k,\cdot)), DG(B_j(k,\cdot))) = d/\alpha$ if $\wedge_{h \in [D]}(\neg A_i(k,h) \vee \neg B_j(k,h)) = True$ and $\leq (d/\alpha) - 1$ otherwise. Since we have to take maximum between the cases when we match the symbols c_k and when we don't match c_k, we get the required equalities.

The outer construction

In the remainder of the proof we construct the final sequences x and y with the promised properties. The sequence x is a concatenation of the inner gadgets $IG(A_i)$ with some additional symbols. The sequence y is a concatenation of the inner gadgets $IG(B_j)$ with some additional symbols. Each inner gadget $IG(B_j)$ appears twice in the sequence y.

We define integer values $v_0 < v_1 < v_2 < v_3$ as follows. We set $v_0 := T'$ (see the definition of the inner gadgets for T'), $v_1 := (1+\delta)T'$, $v_2 := 10v_1$, $v_3 := 100v_1$. For the simplicity of the notation, we will write A_i instead of $IG(A_i)$ and B_j instead of $IG(B_j)$. It will be clear

from the context whether we refer to A_i (B_j, resp.) or to $IG(A_i)$ ($IG(B_j)$, resp.). We define the sequence x:

$$x := \left(\bigcirc_{i=1}^{3n} 2\right) \circ \left(\bigcirc_{i=1}^{n} (0\ A_i\ 1)\right) \circ \left(\bigcirc_{i=1}^{3n} 2\right).$$

We define the sequence y:

$$y := \left(\bigcirc_{j=1}^{n} (2\ 0\ B_j\ 1)\right) \circ \left(\bigcirc_{j=1}^{n} (2\ 0\ B_j\ 1)\right) \circ 2.$$

We set the weight of symbols 0, 1 and 2 as follows: $w(2) := v_2$ and $w(0) = w(1) := v_3$.

We have two goals. First, we want to show that if there are many satisfying assignments (each assignment is a pair of A_i and B_j), then the LCS score between x and y is large: $LCS(x, y) \geq (1 + (\delta/10^5))T$. T is some fixed value that we will define later. Second, if there are no satisfying assignments, then the LCS score is small: $LCS(x, y) \leq T$. We achieve these two goals via the next two lemmas.

▶ **Lemma 21.** *If there are many satisfying assignment (see Definition 1), then*

$$LCS(x, y) \geq (n + 2)v_2 + 2nv_3 + 0.99nv_1 + 0.01nv_0 =: T''.$$

Proof. We will exhibit n different alignments between x and y and we will show that at least one of them achieves the LCS score T''. This gives the lower bound on $LCS(x, y)$.

For $k = 1, \ldots, n$ we write

$$y = \left(\bigcirc_{j=1}^{k-1} (2\ 0\ B_j\ 1)\right) \circ 2 \circ s_k \circ \left(\bigcirc_{j=k}^{n} (2\ 0\ B_j\ 1)\right) \circ 2,$$

where

$$s_k := (0\ B_k\ 1) \circ \left(\bigcirc_{j=k+1}^{n} (2\ 0\ B_j\ 1)\right) \circ \left(\bigcirc_{j=1}^{k-1} (2\ 0\ B_j\ 1)\right).$$

Clearly,

$$\begin{aligned} LCS(x, y) \geq & LCS\left(\bigcirc_{i=1}^{3n} 2, \left(\bigcirc_{j=1}^{k-1} (2\ 0\ B_j\ 1)\right) \circ 2\right) \\ & + LCS\left(\bigcirc_{i=1}^{n} (0\ A_i\ 1), s_k\right) \\ & + LCS\left(\bigcirc_{i=1}^{3n} 2, \left(\bigcirc_{j=k}^{n} (2\ 0\ B_j\ 1)\right) \circ 2\right). \end{aligned}$$

The total contribution of the first and the third term on the r.h.s. is $(n + 2)v_2$ because only symbols 2 can contribute to the LCS score and there are $n + 2$ symbols 2. For the middle term we align inner gadgets in pairs and match all symbols 0 and 1. We get the lower bound

$$LCS(x, y) \geq (n + 2)v_2 + 2nv_3 + \sum_{i=1}^{n} LCS(A_i, B_{j_k(i)}),$$

$$where j_k(i) := \begin{cases} i + k - 1 & \text{if } i \leq n + 1 - k, \\ i + k - 1 - n & \text{otherwise.} \end{cases}$$

By averaging the r.h.s. over all $k = 1, \ldots, n$, we get

$$\begin{aligned} LCS(x, y) &\geq \frac{1}{n} \sum_{k=1}^{n} \left((n + 2)v_2 + 2nv_3 + \sum_{i=1}^{n} LCS(A_i, B_{j_k(i)}) \right) \\ &= (n + 2)v_2 + 2nv_3 + \frac{1}{n} \sum_{i,k=1}^{n} LCS(A_i, B_{j_k(i)}) \\ &= (n + 2)v_2 + 2nv_3 + \frac{1}{n} \sum_{i,j=1}^{n} LCS(A_i, B_j) \\ &\geq (n + 2)v_2 + 2nv_3 + 0.99nv_1 + 0.01nv_0, \end{aligned}$$

where in the last inequality we use the fact that there are many satisfying assignments. This finishes the proof of the lemma. ◀

▶ **Lemma 22.** *If there are no satisfying assignments, then*

$$LCS(x,y) \leq (n+2)v_2 + 2nv_3 + nv_0 =: T.$$

Proof. We start with the intuition behind the analysis.

Intuition

We saw in the proof of Lemma 21 that there is an alignment that achieves a large LCS score. In the alignment we match the n inner gadgets from the first sequence x with an n consecutive inner gadgets from the second sequence y in pairs. We want to claim that in an optimal alignment, we will do the same: map the n inner gadgets from the first sequence with an n consecutive inner gadgets from the second sequence in pairs. Intuitively, this is because of the following three reasons:

- We don't want to choose less than n inner gadgets from the second sequence because otherwise we can't match all symbols 0 and 1 from the first sequence with their counterparts (symbols 0 and 1 have the largest weight - we loose a lot by not matching them).
- We don't want to choose more than n inner gadgets from the second sequence because otherwise we have fewer symbols 2 from the second sequence to be matched with their counterparts. Symbols 2 have smaller weight than symbols 0 and 1 but still we loose a lot by not matching them.
- Finally, if we choose n inner gadgets from the second sequence we want to match them in pairs. If we don't do that, we can't match all symbols 0 and 1 which is again expensive.

We proceed to formalize the intuition.

Sequence x starts with $3n$ copies of symbol 2. Suppose that some of those symbols are matched. W.l.o.g. the matched symbols from a suffix of $\bigcirc_{i=1}^{3n} 2$. W.l.o.g. the last symbol of $\bigcirc_{i=1}^{3n} 2$ is matched. If this is not the case we can match it with the first symbol of y and this can only increase LCS. Consider the symbol 2 from y that is matched to the last symbol 2 from $\bigcirc_{i=1}^{3n} 2$. Consider the symbol to the right of the symbol 2 in y. It is 0. Let s be its position in y. W.l.o.g. this symbol 0 is matched to the first symbol 0 from x. If this is not so, we can make this match and this can't decrease the LCS score. Analogously we can argue that the last symbol 1 from x is matched to a symbol 1 in y. Let t be the location of the symbol 1 in y. Let x' be the substring of x that is to the right of the first symbol 0 in x and to the left of the last symbol 1 in x. Let y' be the substring of y that is to the right of the symbol 0 at location s in y and to the left of the symbol 1 at location t in y. We write $x = x_1 x' x_2$ and $y = y_1 y' y_2$. We can upper bound LCS if we range over all such partitions of x and y:

$$LCS(x,y) \leq \max_{\substack{x=x_1x'x_2 \\ y=y_1y'y_2}} LCS(x_1,y_1) + LCS(x',y') + LCS(x_2,y_2). \tag{1}$$

Let $m \geq 1$ denote the number of inner gadgets in y'.

▶ **Claim 23.**

$$LCS(x_1,y_1) + LCS(x_2,y_2) \leq 2v_3 + (2n+1-(m-1))v_2.$$

Proof. The total number of symbols 2 in y_1 and y_2 is $2n+1-(m-1)$. The total contribution from all symbols 2 is upper bounded by $(2n+1-(m-1))v_2$. We can also match symbol 0 in x_1 and symbol 1 in x_2. This upper bounds the total contribution from symbols 0 and 1 by $2v_3$. There are no other symbols that we can match. The claim follows. ◀

It remains to give an upper bound on $LCS(x', y')$. Consider any symbol 0 in x'. Its *neighbour* is the symbol 1 to the left of it. Similarly, for any symbol 1 in x' the neighbour is the symbol 0 to the right of it. For any symbol 0 in y' the neighbour is the first symbol 1 to the left of it. For any symbol 1 in y' the neighbour is the first symbol 0 to the right of it. For any two symbols 0 that are matched between x' and y', their neighbours (symbols 1) form a match too. If this does not true, we match the symbols 1 and this can only increase the LCS score. Similarly, for any two symbols 1 that are matched, their neighbours form a match too. Let $M \geq 0$ denote the number of pairs of matched symbols 0 and 1. This allows us to upper bound the total contribution from symbols 0 and 1 to $LCS(x', y')$ by $S := 2Mv_2$. The M pairs of matched symbols split the sequence x into $M+1$ maximal substrings $r_1, \ldots, r_{M+1}$. In each one of the $M+1$ substrings s_i does not contain a symbol 0 or 1 that is matched. Similarly, we split y' into $M+1$ maximal substrings $p_1, \ldots, p_{M+1}$ so that each p_i does not contain a symbol 0 or 1 that is matched. Symbols in r_i can only be matched to symbols in p_i. The only symbols that can be matched from r_i with symbols from p_i come from the inner gadgets by the definition of r_i and p_i. Let $d_i \geq 1$ denote the number of the inner gadgets in r_i and $l_i \geq 1$ denote the number of the inner gadgets in p_i. Clearly, $\sum_{i=1}^{M+1} d_i = n$ and $\sum_{i=1}^{M+1} l_i = m$. We claim that $LCS(r_i, p_i) \leq (d_i + l_i - 1)v_0$. Since the pairs of matched symbols can't cross, we can easily check that the total number of pairs of inner gadgets that can have a match is upper bounded by $d_i + l_i - 1$. Because there are no satisfying assignments, the upper bound $LCS(r_i, p_i) \leq (d_i + l_i - 1)v_0$ follows. From all this we have

$$LCS(x', y') \leq S + \sum_{i=1}^{M+1} LCS(r_i, p_i) \leq 2Mv_2 + \sum_{i=1}^{M+1} (d_i + l_i - 1)v_0.$$

We combine this with the equalities $\sum_{i=1}^{M+1} d_i = n$ and $\sum_{i=1}^{M+1} l_i = m$ and get the following Claim.

▶ **Claim 24.**

$$LCS(x', y') \leq 2Mv_3 + (n+m)v_0 - (M+1)v_0.$$

We combine (1) with Claims 23 and 24 and get the following upper bound:

$$\begin{aligned} LCS(x, y) \leq & 2v_2 + n(2v_2 + v_0) \\ & + (M+1)(2v_3 - v_0) - m(v_2 - v_0). \end{aligned}$$

From $\sum_{i=1}^{M+1} d_i = n$ and $\sum_{i=1}^{M+1} l_i = m$ we get that $M \leq \min(n, m) - 1$. As we increase M, the r.h.s. of the upper bound only increases. We choose $M = \min(n, m) - 1$. Consider two cases.

- $m \geq n$. We have $M = n - 1$ and $LCS(x, y) \leq v_2(2n+2) + 2nv_3 - m(v_2 - v_0) \leq T$.
- $m \leq n$. We have $M = m - 1$ and $LCS(x, y) \leq v_2(2n+2) + nv_0 + m(2v_3 - v_2) \leq T$. ◀

From the above Lemmas 21 and 22 we have that $LCS(x, y) \geq T''$ if there are many satisfying assignment and $LCS(x, y) \leq T$ if there are no satisfying assignments. From the definition of values v_0, v_1, v_2, v_3 (in particular, $v_1 = (1+\delta)v_0$), we can easily conclude that $T'' \geq (1 + (\delta/10^5))T$ which gives the properties of x and y that we need.

Harder variants of the Block Disjoint Pairs problem

In the paragraph "Construction of the disjointness gadgets", we do the following construction. Given two vectors $z^k := A_i(k,\cdot), w^k := B_j(k,\cdot) \in \{0,1\}^{d \log N}$, we construct sequences $DG(z^k)$ and $DG(w^k)$ such that LCS between them is $LCS(z^k, w^k) = d/\alpha$ if the vectors are orthogonal and $LCS(z^k, w^k) = (d/\alpha) - 1$ otherwise. We split the vector z^k into d/α shorter vectors $z^k_1, \ldots, z^k_{d/\alpha} \in \{0,1\}^{\alpha \log N}$. Similarly, we split the vector w^k into d/α shorter vectors $w^k_1, \ldots, z^k_{d/\alpha} \in \{0,1\}^{\alpha \log N}$. We construct $DG(z^k)$ by replacing each shorter vector z^k_t by a symbol corresponding to it (indexed by the $\alpha \log N$ binary values) and its position. We construct $DG(w^k)$ by replacing each shorter vector w^k_t by a sequence of symbols corresponding to all vectors that are orthogonal to w^k_t. This implies that we have a large LCS score if there are many orthogonal pairs z^k_t, w^k_t of short vectors. Instead of replacing w^k_t by a sequence of symbols corresponding to all orthogonal vectors, we can take an arbitrary function $f^k_t : \{0,1\}^{2\alpha \log N} \to \{0,1\}$ and replace w^k_t by a sequence of symbols corresponding to all vectors $u \in \{0,1\}^{\alpha \log N}$ such that $f^k_t(u, w^k_t) = 1$. We recover the orthogonality constraint by choosing functions f^k_t that evaluates to 1 iff the two vectors are orthogonal. For arbitrary functions $f^k_1, \ldots, f^k_{d/\alpha} : \{0,1\}^{2\alpha \log N} \to \{0,1\}$, we get that $LCS(z^k, w^k) = d/\alpha$ if $f^k_1(z^k_1, w^k_1) = \ldots = f^k_{d/\alpha}(z^k_{d/\alpha}, w^k_{d/\alpha}) = 1$ and $LCS(z^k, w^k) = (d/\alpha) - 1$ otherwise. Clearly, the new version of Block Disjoint Pairs problem is harder to solve than the one restricted to the orthogonality constraints.

To further increase the hardness of the Block Disjoint Pairs problem we can define functions $g^k : \{0,1\}^{d/\alpha} \to \{0,1\}$ and require that $LCS(z^k, w^k) = q$ if

$$g^k(f^k_1(z^k_1, w^k_1), \ldots, f^k_{d/\alpha}(z^k_{d/\alpha}, w^k_{d/\alpha})) = 1$$

and $LCS(z^k, w^k) = q' < q$ otherwise (for some constants q and q'). Notice that previously all functions g^k are AND functions. This modification requires that the gap $(q/q') - 1$ is at least a constant (we have a constant gap for the AND function) and that the functions g^k can be efficiently simulated with LCS.

4.1 Hardness for Approximate Binary LCS and Edit Distance

The results in this section follow from simple observations over [5] that are easy to make with our framework in mind.

We refer the reader to [5] for the definition and background on Branching Programs.

▶ **Theorem 25** (Theorem 2 in [5]). *There is a reduction from SAT on nondeterministic branching programs on m variables, length T, and width W, to an instance of Edit-Distance or LCS on two binary sequences x and y of length $n = 2^{m/2} \cdot T^{O(\log W)}$, and the reduction runs in $O(n)$ time.*

We need additional properties of the reduction from Theorem 25.

▶ **Claim 26.** *Let P be the Branching Program that we want to reduce.*

If we reduce Branching Program P to LCS problem then we have the following two properties:

- *If P is not satisfiable, then $LCS(x,y) \leq C$ for some integer constant $C = C(m,T,W) \leq n$.*
- *If at least half of the assignments satisfy the Branching Program P, then $LCS(x,y) \geq C + (2^{m/2}/2)$.*

If we reduce Branching Program P to Edit-Distance problem then we have the following two properties:

- *If P is not satisfiable, then $Edit(x,y) \geq C$ for some integer constant $C = C(m,T,W) \leq n$.*

- *If at least half of the assignments satisfy the Branching Program P, then $Edit(x, y) \leq C - (2^{m/2}/2)$.*

Proof. The proof follows from the proof of Claim 9 in [5].

Consider the case when P is not satisfiable. The proof does not change - we show that LCS is upper bounded and Edit-Distance is lower bounded by some fixed quantity C.

Consider the case when P is satisfied by at least half of the assignments. In the proof of Claim 9 the authors choose an integer Δ such that the corresponding alignment pairs up two gadgets that form a satisfying assignment to the Branching Program P. When there are many satisfying assignments (at least half), we can show that there is an integer such in the corresponding alignment at least half of the assignments are satisfying. By the properties of the gadgets constructed in [5], we get the required lower bound on LCS and the required upper bound on Edit-Distance. ◀

Theorem 25 and Claim 26 combined give the following theorem.

▶ **Theorem 27.** *Suppose we have a $(1 + \delta)$ approximation algorithm for Edit-Distance or LCS with $\delta = o(1/T^{O(\log W)})$ that runs in $f(n) = f(2^{m/2} \cdot T^{O(\log W)})$ deterministic time for some function f. Then in time $f(2^{m/2} \cdot T^{O(\log W)})$ we can decide if a Branching program on m variables, length T and width W is not satisfiable or at last half of the assignments are satisfying.*

From the discussion in [5] on the connection between BPs and NC circuits, a lower bound for NC^1 follows. Namely, $1 + 1/\text{poly} \log n$ approximation algorithm implies that there exists $f \in \mathsf{E}^{\mathsf{NP}}$ such that $f \notin NC^1$.

Acknowledgments. We thank Piotr Indyk, Michael P. Kim, Dana Moshkovitz, Virginia Vassilevska Williams, and Ryan Williams for helpful discussions on this work.

References

1 Amir Abboud, Arturs Backurs, Thomas Dueholm Hansen, Virginia Vassilevska Williams, and Or Zamir. Subtree isomorphism revisited. In *Proc. of 27th SODA*, pages 1256–1271, 2016.

2 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Tight Hardness Results for LCS and other Sequence Similarity Measures. In *Proc. of 56th FOCS*, pages 59–78, 2015.

3 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. If the current clique algorithms are optimal, so is valiant's parser. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 98–117. IEEE, 2015.

4 Amir Abboud, Fabrizio Grandoni, and Virginia Vassilevska Williams. Subcubic equivalences between graph centrality problems, APSP and diameter. In *Proc. of 26th SODA*, pages 1681–1697, 2015.

5 Amir Abboud, Thomas Dueholm Hansen, Virginia Vassilevska Williams, and Ryan Williams. Simulating Branching Programs with Edit Distance and Friends or: A Polylog Shaved is a Lower Bound Made. In *STOC'16*, 2016.

6 Amir Abboud and Virginia Vassilevska Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *Proc. of 55th FOCS*, pages 434–443, 2014.

7 Amir Abboud, Virginia Vassilevska Williams, and Joshua R. Wang. Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. In *Proc. of 27th SODA*, pages 377–391, 2016.

8 Amir Abboud, Virginia Vassilevska Williams, and Oren Weimann. Consequences of faster sequence alignment. In *Proc. of 41st ICALP*, pages 39–51, 2014.

9 Amir Abboud, Virginia Vassilevska Williams, and Huacheng Yu. Matching triangles and basing hardness on an extremely popular conjecture. In *Proc. of 47th STOC*, pages 41–50, 2015.

10 Amir Abboud, Ryan Williams, and Huacheng Yu. More applications of the polynomial method to algorithm design. In *Proc. of 26th SODA*, pages 218–230, 2015.

11 Amir Abboud, Virginia Vassilevska Williams, and Joshua Wang. Approximation and fixed parameter subquadratic algorithms for radius and diameter. *arXiv preprint arXiv:1506.01799*, 2015.

12 Josh Alman, Timothy M Chan, and Ryan Williams. Polynomial representations of threshold functions and algorithmic applications. In *to appear at FOCS*, 2016.

13 Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402, 1997.

14 A. Amir, T. M. Chan, M. Lewenstein, and N. Lewenstein. On hardness of jumbled indexing. In *Proc. ICALP*, volume 8572, pages 114–125, 2014.

15 Alexandr Andoni, Robert Krauthgamer, and Krzysztof Onak. Polylogarithmic approximation for edit distance and the asymmetric query complexity. In *FOCS*, pages 377–386, 2010.

16 László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. Bpp has subexponential time simulations unless exptime has publishable proofs. In *Structure in Complexity Theory Conference, 1991., Proceedings of the Sixth Annual*, pages 213–219. IEEE, 1991.

17 Rolf Backofen, Dekel Tsur, Shay Zakov, and Michal Ziv-Ukelson. Sparse rna folding: Time and space efficient algorithms. *Journal of Discrete Algorithms*, 9(1):12–31, 2011.

18 Arturs Backurs, Nishanth Dikkala, and Christos Tzamos. Tight hardness results for maximum weight rectangles. *arXiv preprint arXiv:1602.05837*, 2016.

19 Arturs Backurs and Piotr Indyk. Edit Distance Cannot Be Computed in Strongly Subquadratic Time (unless SETH is false). In *Proc. of 47th STOC*, pages 51–58, 2015.

20 Arturs Backurs and Piotr Indyk. Which regular expression patterns are hard to match? *arXiv preprint arXiv:1511.07070*, 2015.

21 Ziv Bar-Yossef, TS Jayram, Robert Krauthgamer, and Ravi Kumar. Approximating edit distance efficiently. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 550–559. IEEE, 2004.

22 Tuğkan Batu, Funda Ergun, and Cenk Sahinalp. Oblivious string embeddings and edit distance approximations. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 792–801. Society for Industrial and Applied Mathematics, 2006.

23 Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Short pcps verifiable in polylogarithmic time. In *Computational Complexity, 2005. Proceedings. Twentieth Annual IEEE Conference on*, pages 120–134. IEEE, 2005.

24 Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust pcps of proximity, shorter pcps and applications to coding. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 1–10, 2004. doi:10.1145/1007352.1007361.

25 Eli Ben-Sasson and Madhu Sudan. Short pcps with polylog query complexity. *SIAM Journal on Computing*, 38(2):551–607, 2008.

26 Eli Ben-Sasson and Emanuele Viola. Short pcps with projection queries. In *ICALP, Part I*, pages 163–173, 2014.

27 Lasse Bergroth, Harri Hakonen, and Timo Raita. New approximation algorithms for longest common subsequences. In *String Processing and Information Retrieval: A South American Symposium, 1998. Proceedings*, pages 32–40. IEEE, 1998.

28 Lasse Bergroth, Harri Hakonen, and Timo Raita. A survey of longest common subsequence algorithms. In *String Processing and Information Retrieval, 2000. SPIRE 2000. Proceedings. Seventh International Symposium on*, pages 39–48. IEEE, 2000.

29 Mark Braverman, Young Kun-Ko, and Omri Weinstein. Approximating the best nash equilibrium in $n^{o(\log n)}$-time breaks the exponential time hypothesis. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 970–982, 2015. doi:10.1137/1.9781611973730.66.

30 Karl Bringmann. Why walking the dog takes time: Frechet distance has no strongly subquadratic algorithms unless seth fails. In *Proc. of 55th FOCS*, pages 661–670, 2014.

31 Karl Bringmann and Marvin Künnemann. Improved approximation for fréchet distance on c-packed curves matching conditional lower bounds. *CoRR*, abs/1408.1340, 2014. URL: http://arxiv.org/abs/1408.1340.

32 Karl Bringmann and Marvin Kunnemann. Quadratic Conditional Lower Bounds for String Problems and Dynamic Time Warping. In *Proc. of 56th FOCS*, pages 79–97, 2015.

33 Karl Bringmann and Wolfgang Mulzer. Approximability of the Discrete Fréchet Distance. In *Proc. of 31st SoCG*, pages 739–753, 2015.

34 Chris Calabro. A lower bound on the size of series-parallel graphs dense in long paths. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 15, 2008.

35 Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. The complexity of satisfiability of small depth circuits. In *Proc. of 4th IWPEC*, pages 75–85, 2009.

36 Marco L Carmosino, Jiawei Gao, Russell Impagliazzo, Ivan Mihajlin, Ramamohan Paturi, and Stefan Schneider. Nondeterministic extensions of the strong exponential time hypothesis and consequences for non-reducibility. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 261–270. ACM, 2016.

37 Marco L Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Tighter connections between derandomization and circuit lower bounds. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 40. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

38 Diptarka Chakraborty, Elazar Goldenberg, and Michal Koucký. Streaming algorithms for embedding and computing edit distance in the low distance regime. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, 2015.

39 Timothy M Chan and Ryan Williams. Deterministic apsp, orthogonal vectors, and more: Quickly derandomizing razborov-smolensky. In *Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA, 2016.

40 Yi-Jun Chang. Hardness of rna folding problem with four symbols. *arXiv preprint arXiv:1511.04731*, 2015.

41 Krishnendu Chatterjee, Wolfgang Dvořák, Monika Henzinger, and Veronika Loitzenbauer. Model and objective separation with conditional lower bounds: Disjunction is harder than conjunction. *arXiv preprint arXiv:1602.02670*, 2016.

42 Shiri Chechik, Daniel H Larkin, Liam Roditty, Grant Schoenebeck, Robert E Tarjan, and Virginia Vassilevska Williams. Better approximation algorithms for the graph diameter. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1041–1052. SIAM, 2014.

43 F Chin and Chung Keung Poon. Performance analysis of some simple heuristics for computing longest common subsequences. *Algorithmica*, 12(4-5):293–311, 1994.

44 Thomas H.. Cormen, Charles Eric Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*, volume 6. MIT press Cambridge, 2001.

45 Maxime Crochemore, Costas S Iliopoulos, Yoan J Pinzon, and James F Reid. A fast and practical bit-vector algorithm for the longest common subsequence problem. *Information Processing Letters*, 80(6):279–285, 2001.

46 Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström. On problems as hard as cnf-sat. In *Computational Complexity (CCC), 2012 IEEE 27th Annual Conference on*, pages 74–84. IEEE, 2012.

47 Søren Dahlgaard. On the hardness of partially dynamic graph problems and connections to diameter. *arXiv preprint arXiv:1602.06705*, 2016.

48 J Boutet de Monvel. Extensive simulations for longest common subsequences. *The European Physical Journal B-Condensed Matter and Complex Systems*, 7(2):293–308, 1999.

49 Robert C Edgar and Serafim Batzoglou. Multiple sequence alignment. *Current opinion in structural biology*, 16(3):368–373, 2006.

50 Lance Fortnow and Adam R Klivans. Efficient learning algorithms yield circuit lower bounds. *Journal of Computer and System Sciences*, 75(1):27–36, 2009.

51 Yelena Frid and Dan Gusfield. A simple, practical and complete o-time algorithm for rna folding using the four-russians speedup. *Algorithms for Molecular Biology*, 5(1):1, 2010.

52 A. Gajentaan and M. H. Overmars. On a class of $o(n^2)$ problems in computational geometry. *Comput. Geom. Theory Appl.*, 45(4):140–152, 2012.

53 Ofer Grossman and Dana Moshkovitz. Amplification and derandomization without slowdown. *arXiv preprint arXiv:1509.08123*, 2015.

54 Dan Gusfield. *Algorithms on strings, trees and sequences: computer science and computational biology*. Cambridge university press, 1997.

55 Monika Henzinger, Sebastian Krinninger, Danupon Nanongkai, and Thatchaphol Saranurak. Unifying and strengthening hardness for dynamic problems via the online matrix-vector multiplication conjecture. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 21–30. ACM, 2015.

56 Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. In search of an easy witness: Exponential time vs. probabilistic polynomial time. *Journal of Computer and System Sciences*, 65(4):672–694, 2002.

57 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

58 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63:512–530, 2001.

59 Russell Impagliazzo and Avi Wigderson. P= bpp if e requires exponential circuits: Derandomizing the xor lemma. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 220–229. ACM, 1997.

60 Hamid Jahanjou, Eric Miles, and Emanuele Viola. Local reductions. In *Automata, Languages, and Programming*, pages 749–760. Springer, 2015.

61 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004.

62 Richard M Karp and Richard Lipton. Turing machines that take advice. *Enseign. Math*, 28(2):191–209, 1982.

63 Kazutaka Katoh and Daron M Standley. Mafft multiple sequence alignment software version 7: improvements in performance and usability. *Molecular biology and evolution*, 30(4):772–780, 2013.

64 Adam Klivans, Pravesh Kothari, and Igor C Oliveira. Constructing hard functions using learning algorithms. In *Computational Complexity (CCC), 2013 IEEE Conference on*, pages 86–97. IEEE, 2013.

65 Gad M Landau, Eugene W Myers, and Jeanette P Schmidt. Incremental string comparison. *SIAM Journal on Computing*, 27(2):557–582, 1998.

66 William J Masek and Michael S Paterson. A faster algorithm computing string edit distances. *Journal of Computer and System sciences*, 20(1):18–31, 1980.

67 Thilo Mie. Short pcpps verifiable in polylogarithmic time with o (1) queries. *Annals of Mathematics and Artificial Intelligence*, 56(3-4):313–338, 2009.

68 Gonzalo Navarro. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1):31–88, 2001.

69 Rafail Ostrovsky and Yuval Rabani. Low distortion embeddings for edit distance. *Journal of the ACM (JACM)*, 54(5):23, 2007.

70 Mihai Patrascu. Towards polynomial lower bounds for dynamic problems. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 603–610. ACM, 2010.

71 Liam Roditty and Virginia Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Proc. of 45th STOC*, pages 515–524, 2013.

72 Balaram Saha. The dyck language edit distance problem in near-linear time. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 611–620. IEEE, 2014.

73 Barna Saha. Language edit distance and maximum likelihood parsing of stochastic grammars: Faster algorithms and connection to fundamental graph problems. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 118–135. IEEE, 2015.

74 Rajesh Santhanam and Ross Williams. On medium-uniformity and circuit lower bounds. In *Computational Complexity (CCC), 2013 IEEE Conference on*, pages 15–23. IEEE, 2013.

75 Temple F Smith and Michael S Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.

76 Yinglei Song. Time and space efficient algorithms for rna folding with the four-russians technique. *arXiv preprint arXiv:1503.05670*, 2015.

77 Julie D Thompson, Desmond G Higgins, and Toby J Gibson. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, 22(22):4673–4680, 1994.

78 Leslie G Valiant. *Graph-theoretic arguments in low-level complexity*. Springer, 1977.

79 Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *Proc. of 51st FOCS*, pages 645–654, 2010.

80 Balaji Venkatachalam, Dan Gusfield, and Yelena Frid. Faster algorithms for rna-folding using the four-russians method. *Algorithms for Molecular Biology*, 9(1):1, 2014.

81 Emanuele Viola. *On the power of small-depth computation*. Now Publishers Inc, 2009.

82 Ryan Williams. A new algorithm for optimal constraint satisfaction and its implications. In *Automata, Languages and Programming*, pages 1227–1237. Springer, 2004.

83 Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. *SIAM Journal on Computing*, 42(3):1218–1244, 2013.

84 Ryan Williams. Natural proofs versus derandomization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 21–30. ACM, 2013.

85 Ryan Williams. Algorithms for Circuits and Circuits for Algorithms: Connecting the Tractable and Intractable. In *Proceedings of the International Congress of Mathematicians*, 2014. URL: `http://web.stanford.edu/~{}rrwill/ICM-survey.pdf`.

86 Ryan Williams. Nonuniform ACC circuit lower bounds. *J. ACM*, 61(1):2:1–2:32, 2014.

87 Ryan Williams. Strong ETH Breaks With Merlin and Arthur: Short Non-Interactive Proofs of Batch Evaluation. In *CCC'16*, 2016.

Parameterized Property Testing of Functions*

Ramesh Krishnan S. Pallavoor[1], Sofya Raskhodnikova[2], and Nithin Varma[3]

1 **Pennsylvania State University, University Park, USA**
`rxp271@cse.psu.edu`
2 **Pennsylvania State University, University Park, USA**
`sofya@cse.psu.edu`
3 **Pennsylvania State University, University Park, USA**
`nithvarma@psu.edu`

Abstract

We investigate the parameters in terms of which the complexity of sublinear-time algorithms should be expressed. Our goal is to find input parameters that are tailored to the combinatorics of the specific problem being studied and design algorithms that run faster when these parameters are small. This direction enables us to surpass the (worst-case) lower bounds, expressed in terms of the input size, for several problems. Our aim is to develop a similar level of understanding of the complexity of sublinear-time algorithms to the one that was enabled by research in parameterized complexity for classical algorithms.

Specifically, we focus on testing properties of functions. By parameterizing the query complexity in terms of the size r of the image of the input function, we obtain testers for monotonicity and convexity of functions of the form $f : [n] \to \mathbb{R}$ with query complexity $O(\log r)$, with no dependence on n. The result for monotonicity circumvents the $\Omega(\log n)$ lower bound by Fischer (Inf. Comput., 2004) for this problem. We present several other parameterized testers, providing compelling evidence that expressing the query complexity of property testers in terms of the input size is not always the best choice.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Sublinear algorithms, property testing, parameterization, monotonicity, convexity.

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.12

1 Introduction

In this paper, we set out to investigate the parameters in terms of which the complexity of sublinear-time algorithms should be expressed. Our goal is to find input parameters that are tailored to the combinatorics of the specific problem being studied and design algorithms that run faster when these parameters are small. This direction could enable one to surpass the (worst-case) lower bounds on the problem complexity that are usually expressed in terms of the input size. The spirit of our study is similar to that in the field of parameterized complexity. In parameterized complexity, the focus is on expressing the complexity of problems as a function of one or more input parameters in order to obtain a fine-grained complexity classification, for example, of NP-hard problems. Our aim is to

* This work was supported by NSF grant CCF-1422975; the third author was also supported by Pennsylvania State University College of Engineering Fellowship and Pennsylvania State University Graduate Fellowship.


© Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and Nithin Varma;
licensed under Creative Commons License CC-BY

develop a similar level of understanding of the complexity of sublinear-time algorithms to the one that was enabled by research in parameterized complexity for classical algorithms.

We focus our study on the framework of property testing, introduced by Goldreich et al. [28] and Rubinfeld and Sudan [40]. In property testing, an algorithm (an ε-tester) for property $\mathcal{P}$, where $\mathcal{P}$ is viewed as a class of functions, is given a parameter $\varepsilon \in (0,1)$ as input and has oracle access to a function f. The tester has to accept with probability at least $2/3$ if f belongs to the class $\mathcal{P}$, and reject with probability at least $2/3$ if f is *ε-far* from $\mathcal{P}$, that is, differs from every function in $\mathcal{P}$ on at least an ε fraction of function values. In the context of property testing of functions, the query complexity of a tester is usually expressed in terms of ε and the size of the domain of the input function. This works well for properties whose query complexity depends only on the proximity parameter ε. However, for other properties, it is not clear whether the domain size is the right parameter to express their testing complexity.

Consider, for example, the widely studied problem of testing monotonicity of real-valued functions (see, e.g., [27, 22, 23, 36, 26, 24, 31, 1, 32, 2, 11, 10, 13, 16, 12, 9, 17, 18, 15, 20, 19, 35, 4, 5, 21], and recent surveys [38, 14]). For functions over a discrete domain $[n]$ (also called the line), monotonicity testing is equivalent to testing sortedness of arrays. Algorithms for sortedness testing have found use, for instance, in determining the "state of sortedness" of relational databases [6], where the testing step is performed to decide on the sorting algorithms to be run on the database. The complexity of sortedness testing (for constant ε) is $\Theta(\sqrt{n})$ if the tester is only allowed to make independent and uniformly random queries [26]; it is $\Theta(\log n)$ if the tester is allowed to make arbitrary queries [23, 24].

From the above discussion, it might appear that one cannot make any more improvements to the complexity of monotonicity testing over $[n]$. However, we argue that this is the case only when the complexity of the problem is parameterized in terms of n, the domain size.

In this work, we ask whether better monotonicity testers can be designed by parameterizing the query complexity in terms of the size of the image of the input function. The starting point for our investigation is the folklore result that, for ε-testing monotonicity of Boolean functions over $[n]$, only $O(1/\varepsilon)$ queries suffice. A slightly more general corollary of this result is that monotonicity of functions over $[n]$ with image size at most 2 can be ε-tested with only $O(1/\varepsilon)$ queries. The only bound for monotonicity testing (over $[n]$) that is expressed in terms of the image size r of the input function is the bound of $\Omega(\log r)$ for nonadaptive[1] testers due to Blais et al. [12]. We design an ε-tester for monotonicity of functions over $[n]$ with query complexity $O((\log r)/\varepsilon)$, where r is an upper bound on the size of the image of the input function. This result circumvents Fischer's lower bound of $\Omega(\log n)$ for this problem by focusing on a different parameter for measuring query complexity.

The size of the image is one of the natural parameters in terms of which one can express the complexity of property testing algorithms. In this work, we show that there are several testing problems for which parameterizing the complexity in terms of the image size works well. Another example where parameterization has helped in the design of efficient testers is the work of Jha and Raskhodnikova [34] on Lipschitz testing, even though they do not view their results from this angle. The complexity of their testers is expressed in terms of the *image diameter*. The *image diameter* of a function $f : \mathcal{D} \mapsto \mathbb{R}$ is $\max_{x,y\in\mathcal{D}} |f(x) - f(y)|$. In many situations, the image diameter is much smaller than the domain size. We believe that all this evidence is compelling enough to make one rethink the way in which the complexity

[1] Testers whose queries do not depend on the answers to previous queries are called *nonadaptive*; general testers that do not satisfy this requirement are *adaptive*.

of sublinear-time algorithms is expressed. Our paper is a first step towards formalizing this notion and finding what we think are the right parameters to express the complexity of some central problems in sublinear-time algorithms.

1.1 Parameters and Properties Studied in this Work

We study the dependence of complexity of monotonicity and convexity testers on the image size of the input functions. The image of a function is defined as follows.

▶ **Definition 1.1** (Image of a function)**.** Let f be a function defined over a finite, discrete domain $\mathcal{D}$. The image of f, denoted $\mathrm{Im}(f)$, is the set $\{f(x) : x \in \mathcal{D}\}$ or, in other words, the set of all values taken by f on points in $\mathcal{D}$.

For the special case, when $\mathcal{D}$ is $[n]$, a function $f : [n] \mapsto \mathbb{R}$ can also be viewed as a real-valued array of length n. Here, $\mathrm{Im}(f)$ is equal to the set of distinct values in the array.

We restrict our attention to real-valued functions defined over the following domains. These are domains for which testing monotonicity and convexity have been studied extensively.

▶ **Definition 1.2** (Hypergrid, Hypercube, Line)**.** For $x \in [n]^d$, let x_i denote the i^{th} coordinate of x. A *hypergrid* is a partial order $([n]^d, \preceq)$ where $x \preceq y$ means that $x_i \leq y_i$ for all $x, y \in [n]^d$ and $i \in [d]$. The partial order $([2]^d, \preceq)$ is called a *hypercube* and the total order $([n], \preceq)$ is called a *line*.

Next, we summarize some of the previous work on testing monotonicity and convexity of real-valued functions.

Monotonicity. A function $f : \mathcal{D} \mapsto \mathbb{R}$ defined over a partial order $(\mathcal{D}, \preceq)$ is *monotone* if $f(x) \leq f(y)$ for all $x, y \in \mathcal{D}$ satisfying $x \preceq y$. Monotonicity is one of the most widely studied properties in the field of property testing [27, 22, 23, 36, 26, 24, 31, 1, 32, 2, 11, 10, 13, 16, 12, 9, 17, 18, 15, 20, 19, 35, 4, 5, 21]. The complexity of ε-testing monotonicity of functions of the form $f : [n]^d \mapsto \mathbb{R}$ is $\Theta\left(\frac{d \log n}{\varepsilon}\right)$ [16, 17]. For the special case of the line, the testing complexity is $\Theta\left(\frac{\log n}{\varepsilon}\right)$ [23, 24]. For functions defined over general poset domains $\mathcal{D}$, the complexity of monotonicity testing is $O\left(\sqrt{|\mathcal{D}|/\varepsilon}\right)$ [26].

Convexity. For a convex set $\mathcal{D}$, a function $f : \mathcal{D} \mapsto \mathbb{R}$ is convex if $\forall x, y \in \mathcal{D}$ and $t \in [0, 1]$, $f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$. For real-valued functions over $[n]$, convexity can be ε-tested using $O(\frac{\log n}{\varepsilon})$ queries [37]. This bound is tight (for constant ε) for nonadaptive testers [12].

1.2 Our Results

In this section, we describe the key technical contributions of our work. We design efficient testers for monotonicity over various hypergrid domains and convexity over the line. For monotonicity of functions over the line $[n]$, which is equivalent to the property of sortedness of arrays of length n, we design efficient testers under two different models of input access: (i) query access and (ii) uniform samples. Our testers are given an upper bound r on the image size of the input function, and their complexity is parameterized in terms of r.

Sortedness testing. We present our tester for sortedness of n-element arrays (monotonicity over the line $[n]$) in Section 3. The complexity of our tester is independent of n. Our tester has 1-sided error, that is, it always accepts a function with the property. (In contrast, the general tester is said to have 2-sided error.) We prove the following theorem.

▶ **Theorem 1.3.** *There exists a 1-sided error ε-tester making $O\left(\frac{\log r}{\varepsilon}\right)$ queries to test sortedness of arrays with at most r distinct values.*

An important ingredient in our sortedness tester is a nearly optimal nonadaptive tester for this task, presented in Section 2. Its performance is summarized in the next theorem.

▶ **Theorem 1.4.** *There exists a nonadaptive, 1-sided error ε-tester making $O\left(\frac{1}{\varepsilon}\log\frac{r}{\varepsilon}\right)$ queries to test sortedness of arrays with at most r distinct values.*

The query complexity of our nonadaptive tester matches (for constant ε) the $\Omega(\log r)$ lower bound for nonadaptive sortedness testers in [12]. Note that for $r \geq 1/\varepsilon$, the complexity of the nonadaptive tester is $O\left(\frac{\log r}{\varepsilon}\right)$. The tester that we design to prove Theorem 1.3 runs the nonadaptive tester for $r \geq 1/\varepsilon$ and a different (adaptive) tester, presented in Section 3, for $r < 1/\varepsilon$.

Uniform sortedness testing. The work that defined property testing [28], in addition to the model with oracle access to the input, also considered testers that are allowed access to function values only at points sampled uniformly and independently at random from the domain. This model of property testing, known as *uniform* or *sample-based* testing, was further studied by Goldreich and Ron [30], Fischer et al. [25], Berman et al. [8] and Berman et al. [7]. The query complexity of ε-testing sortedness of n-element arrays (for constant ε) using only uniformly and independently drawn samples is $\Theta(\sqrt{n})$ [26]. We design optimal (up to the dependence on ε) uniform testers whose query complexity is parameterized in terms of the number or distinct elements in the input arrays. These results can be found in Sections 5 and 6.

▶ **Theorem 1.5.** *There exists a 1-sided error ε-tester that makes $O(\sqrt{r}/\varepsilon)$ uniform and independent queries to test sortedness of arrays with at most r distinct values.*

▶ **Theorem 1.6.** *Testing sortedness of arrays with values in $[r]$ requires $\Omega(\sqrt{r})$ uniform queries, even with 2-sided error.*

Monotonicity testing over hypergrids. We present our tester for monotonicity of real-valued functions over hypergrid domains in Section 4 and prove the following theorem.

▶ **Theorem 1.7.** *There exists a 1-sided error ε-tester that makes $O\left(\frac{d}{\varepsilon}\log\frac{d}{\varepsilon}\log r\right)$ queries to test monotonicity of real-valued functions $f : [n]^d \mapsto \mathbb{R}$ over the hypergrid domain, where $|Im(f)| \leq r$.*

Note that our tester has a better complexity (up to log factors) than the optimal tester for monotonicity of real-valued functions over the hypergrid domains that makes $O\left(\frac{d\log n}{\varepsilon}\right)$ queries [16] for small r. Parameterizing the complexity of testing in terms of the image size of the functions being tested is what enables us to bypass the $\Omega\left(\frac{d\log n}{\varepsilon}\right)$ lower bound for monotonicity testing of functions over hypergrid domains in [17].

Convexity testing over the line. Finally, in Section 7, we give a nonadaptive convexity tester for real-valued functions over the line and prove the following theorem.

▶ **Theorem 1.8.** *There exists a nonadaptive, 1-sided error ε-tester for convexity of functions $f : [n] \mapsto \mathbb{R}$ that takes an integer $r \geq |Im(f)|$ as input and makes $O(1/\varepsilon)$ queries when $r < \varepsilon n/3$ and $O\left(\frac{\log(r/\varepsilon)}{\varepsilon}\right)$ queries otherwise.*

Recall that for real-valued functions over $[n]$, the complexity of (nonadaptively) ε-testing convexity (for constant ε) is $\Theta(\log n)$. Contrary to this, our tester makes only a constant number of queries when the image size of the function is small.

1.3 Related Work

A related concept of parameterized testability of graph properties was studied by Iwama and Yoshida [33]. The focus of their work was to design efficient algorithms for the property testing variants of several NP-hard decision problems on graphs, by expressing their complexity in terms of parameters that have been successfully used in the literature on parameterized algorithms. In most of the cases, the parameters that they used are NP-hard to compute. In contrast, our goal is to determine the right input parameters in terms of which to express the complexity of property testers and, more generally, sublinear-time algorithms. The parameters we use are often easy to compute or estimate and, in many situations, can be assumed to be given to the algorithm. We also believe that the parameters that we use are tied to the intrinsic combinatorial structure of the properties and give insights into complexity of testing them.

2 The Nonadaptive Sortedness Tester

In this section, we describe a nonadaptive, 1-sided error ε-tester for sortedness of arrays containing at most r distinct values and prove Theorem 1.4. Our tester (Algorithm 1) uses a proximity oblivious tester (POT) for sortedness as a subroutine.

▶ **Definition 2.1** (POT, Goldreich and Ron [29])**.** A *proximity oblivious tester* for a property $\mathcal{P}$ is an algorithm that has oracle access to a function f and

1. always accepts if $f \in \mathcal{P}$;
2. rejects with probability at least $\text{dist}(f, \mathcal{P})$ if $f \notin \mathcal{P}$, where $\text{dist}(f, \mathcal{P})$ is the minimum fraction of values in f that needs to be changed, so that $f \in \mathcal{P}$.

Observe that a POT for $\mathcal{P}$ can be repeated $O(1/\varepsilon)$ times to obtain a 1-sided error ε-tester for $\mathcal{P}$. We note that Definition 2.1 is a special case of the definition of POT in [29]. Specifically, Goldreich and Ron [29] allow the rejection probability of a POT to be a non-decreasing function of $\text{dist}(f, \mathcal{P})$. However, the special case in Definition 2.1 is sufficient for our purposes.

We now give an overview of Algorithm 1. It runs for $O(1/\varepsilon)$ iterations. In each iteration, it first runs a POT for sortedness on a subarray B of the input array A consisting of $1 + 2r/\varepsilon$ (nearly) equally spaced indices. Next, it picks an index $i \in [n]$ uniformly at random. It compares $A[i]$ with the array values of the indices closest to i that were included in B. Algorithm 1 rejects if either of these steps finds elements out of order.

At least three distinct POTs for sortedness of arrays with $O(\log n)$ query complexity are known [23, 10, 16]. We can use any of them in Algorithm 1. Note that Algorithm 1 is not proximity oblivious itself, as it uses the proximity parameter ε to determine its queries. For simplicity, we assume throughout that $2r/\varepsilon$ is an integer that divides n.

Algorithm 1: The Nonadaptive Sortedness Tester

input: query access to an array A of size n, an upper bound r on the number of distinct values in A, and a distance parameter $\varepsilon \in (0,1)$.

1 Let B be the subarray of A consisting of the indices $1, \frac{\varepsilon n}{2r}, \frac{2\varepsilon n}{2r}, \ldots, \left(\frac{2r}{\varepsilon}-1\right)\frac{\varepsilon n}{2r}, n$.
`// No need to explicitly construct` B.
2 **repeat** $\left\lceil \frac{8}{\varepsilon} \right\rceil$ **times**
3 Run a POT for sortedness of arrays (e.g., from [23], [10] or [16]) on B and **reject** if it rejects.
4 Query an index i from A uniformly at random.
5 Set $k = \left\lfloor \frac{2ri}{\varepsilon n} \right\rfloor + 1$. `// Note that` $B[k] = A\left[\frac{(k-1)\varepsilon n}{2r}\right]$ `and` $B[k+1] = A\left[\frac{k\varepsilon n}{2r}\right]$.
6 Query $B[k]$ and $B[k+1]$.
7 **Reject** if $(B[k], A[i], B[k+1])$ is not in sorted order.
8 **Accept**.

Proof of Theorem 1.4. We prove that Algorithm 1 is a nonadaptive, 1-sided error ε-tester making $O\left(\frac{1}{\varepsilon}\log\frac{r}{\varepsilon}\right)$ queries to test sortedness of arrays with at most r distinct values. Algorithm 1 is nonadaptive, since its queries can be chosen in advance. It has 1-sided error as it always accepts sorted arrays. Lemmas 2.2 and 2.3 complete the proof of Theorem 1.4. ◀

▶ **Lemma 2.2.** *Algorithm 1 makes $O\left(\frac{1}{\varepsilon}\log\frac{r}{\varepsilon}\right)$ queries.*

Proof. The query complexity of Step 3 is $O(\log|B|) = O(\log(r/\varepsilon))$. Steps 4-7 make a constant number of queries. Steps 3-7 are executed $O(1/\varepsilon)$ times. Hence, the overall query complexity of the tester is $O\left(\frac{1}{\varepsilon}\log\frac{r}{\varepsilon}\right)$. ◀

Recall that an array is *ε-far* from sorted if at least an ε fraction of elements need to be modified to make it sorted; otherwise, it is *ε-close* to sorted.

▶ **Lemma 2.3.** *Algorithm 1, with probability at least $2/3$, rejects every array that has at most r distinct values and is ε-far from sorted.*

Proof. Consider an array A that has at most r distinct values and is ε-far from sorted. Let B be the subarray of A as defined in Step 1 of Algorithm 1. If B is $\frac{\varepsilon}{7}$-far from sorted, then by the definition of POT for sortedness, Step 3 of our tester rejects with probability at least $\frac{\varepsilon}{7}$ in each iteration. In the rest of the proof, we consider the case when B is $\frac{\varepsilon}{7}$-close to sorted.

▶ **Claim 2.4.** If B is $\frac{\varepsilon}{7}$-close to sorted, then Steps 4-7 reject with probability at least $\frac{\varepsilon}{7}$ in each iteration.

Proof. The subarray B consists of $1 + 2r/\varepsilon$ (nearly) equally spaced indices, which partition A into $2r/\varepsilon$ intervals of nearly the same size. Let $\mathcal{I} = \{I_1, I_2, \ldots, I_{2r/\varepsilon}\}$ denote the set of these intervals. Here, I_1 denotes the interval[2] $\left[2..\frac{\varepsilon n}{2r}-1\right]$ and, for $k > 1$, the interval $\left[\frac{(k-1)\varepsilon n}{2r}+1..\frac{k\varepsilon n}{2r}-1\right]$ is denoted by I_k. Note that, by definition, $B[k]$ and $B[k+1]$ denote the values of the elements in A present immediately to the left and right of I_k, respectively.

An interval I_k is *nearly-constant* if $B[k] = B[k+1]$. Let $\mathcal{C}_t$ be the set of arrays with all their values equal to t. Let $A[I_k]$ denote the subarray of A on the indices in I_k. Let $d(I_k)$

[2] We use $[a..b]$ to denote $\{a, a+1, \ldots, b-1, b\}$ for $a, b \in \mathbb{Z}, a < b$.

and $D(I_k)$ denote the fractional and absolute Hamming distance of $A[I_k]$ from the property $\mathcal{C}_{B[k]}$. Note that $d(I_k) = D(I_k)/|I_k|$.

We now prove Claim 2.4 as follows in two steps. First, we show that $\sum_{I_k \in \mathcal{I}'} D(I_k) > \varepsilon n/7$, where $\mathcal{I}' = \{I_k \in \mathcal{I} : I_k \text{ is nearly-constant}\}$. Second, we show that Steps 4-7 of Algorithm 1 reject with probability at least $\sum_{k \in \mathcal{I}'} D(I_k)/n$ in each iteration.

Since B is $\frac{\varepsilon}{7}$-close to sorted, there exists a set S of at most $\varepsilon|B|/7$ indices in B whose values can be changed to make B sorted. Note that, for $r \geq 3$, we have $|S| < r/3$ since $|B| = 1 + 2r/\varepsilon$. Consider the set of intervals E_1 in $\mathcal{I}$ adjacent to at least one index from S. As each index in S is adjacent to at most two intervals, $|E_1| < 2r/3$.

Let E_2 denote the set of intervals in $\mathcal{I} \setminus E_1$ that are not nearly-constant. For all k such that $I_k \in E_2$, we have $B[k] < B[k+1]$. This is so because, if $B[k] > B[k+1]$, then $I_k \in E_1$ and if $B[k] = B[k+1]$, then I_k is nearly-constant. The total number of distinct values taken by the elements belonging to intervals in E_2 is at least $|E_2|$. But A has at most r distinct values, and hence, $|E_2| \leq r$. Consequently, $|E_1 \cup E_2| < \frac{2r}{3} + r = \frac{5r}{3}$.

Consider the subarray A'' of A induced by the indices in the intervals in $\mathcal{I} \setminus (E_1 \cup E_2)$. Let $D_{\mathcal{S}}(A)$ denote the absolute Hamming distance of the array A to the sortedness property. As $D_{\mathcal{S}}(A) \geq \varepsilon n$, we get $D_{\mathcal{S}}(A'') > \varepsilon n - \frac{5r}{3} \cdot \frac{\varepsilon n}{2r} = \frac{\varepsilon n}{6}$. Note that all the intervals in A'' are nearly-constant. Hence, $(\mathcal{I} \setminus (E_1 \cup E_2)) \subseteq \mathcal{I}'$ and, consequently,

$$\sum_{I_k \in \mathcal{I}'} D(I_k) \geq D_{\mathcal{S}}(A'') > \frac{\varepsilon n}{6} > \frac{\varepsilon n}{7}.$$

This completes the first step of the proof.

Consider a nearly-constant interval $I_k \in \mathcal{I}'$ such that $D(I_k) > 0$. As $B[k] = B[k+1]$, there exists $D(I_k)$ elements in I_k whose values are not $B[k]$, i.e.,

$$|\{x \in I_k : A[x] \neq B[k]\}| = D(I_k).$$

Algorithm 1 rejects if it samples an index $x \in I_k$ in Step 4 such that $B[k] = B[k+1]$ (i.e., $I_k \in \mathcal{I}'$) and $A[x] \neq B[k]$. As there are $\sum_{I_k \in \mathcal{I}'} D(I_k)$ such indices in A, Steps 4-7 of Algorithm 1 reject A with probability at least $\sum_{I_k \in \mathcal{I}'} D(I_k)/n$. Since $\sum_{I_k \in \mathcal{I}'} D(I_k) > \varepsilon n/7$, the proof of Claim 2.4 is complete. ◀

Hence, the probability that Steps 3-7 reject in each iteration is at least $\varepsilon/7$. The probability that Algorithm 1 accepts after $\lceil 8/\varepsilon \rceil$ iterations is at most $(1 - \varepsilon/7)^{8/\varepsilon} \leq e^{-8/7} < 1/3$. This completes the proof of Lemma 2.3. ◀

3 The Sortedness Tester with $O\left(\frac{\log r}{\varepsilon}\right)$ Query Complexity

In this section, we describe a 1-sided error ε-tester for sortedness of arrays containing at most r distinct values and prove Theorem 1.3. The tester, described in Algorithm 2, runs the nonadaptive tester (Algorithm 1) described in Section 2 when $r \geq 1/\varepsilon$, and a different procedure, which is described in Algorithm 2, otherwise.

Proof of Theorem 1.3. We prove that Algorithm 2 is a 1-sided error ε-tester making $O\left(\frac{\log r}{\varepsilon}\right)$ queries to test sortedness of arrays with at most r distinct values. When $r \geq 1/\varepsilon$, Algorithm 2 runs Algorithm 1 and outputs its answer. By Theorem 1.4, Algorithm 1 is a 1-sided error ε-tester with query complexity $O\left(\frac{1}{\varepsilon} \log \frac{r}{\varepsilon}\right)$ which is equal to $O\left(\frac{\log r}{\varepsilon}\right)$ as $r \geq 1/\varepsilon$. When $r < 1/\varepsilon$, Algorithm 2 only rejects if it finds array elements out of order, and so, it has 1-sided error. Lemmas 3.1 and 3.2 complete the proof of Theorem 1.3. ◀

Algorithm 2: The Sortedness Tester

input: query access to an array A of size n, an upper bound r on the number of distinct values in A, and distance parameter $\varepsilon \in (0, 1)$.

1 If $r \geq 1/\varepsilon$, run **Algorithm 1** and **return** its answer.
2 If $A[1] > A[n]$, **reject**.
3 Initialize a balanced binary search tree T to contain keys 1 and n.
`// Define` $\mathsf{successor}(i) = \min\{j \in T : j > i\}$; $\mathsf{predecessor}(i) = \max\{j \in T : j < i\}$.
4 **while** $\exists i, j \in T$ *such that* $j = \mathsf{successor}(i)$ *and* $|i - j| > \frac{\varepsilon n}{2r}$ *and* $A[i] < A[j]$ **do**
5 Set $m = \left\lfloor \frac{i+j}{2} \right\rfloor$ and query $A[m]$.
6 **if** $A[i] \leq A[m] \leq A[j]$ **then** insert m into T **else reject**.
7 **if** $i > 1$ *and* $A[\mathsf{predecessor}(i)] = A[i] = A[m]$ **then**
8 Delete i from T.
9 **if** $j < n$ *and* $A[m] = A[j] = A[\mathsf{successor}(j)]$ **then**
10 Delete j from T.
11 **repeat** $\left\lceil \frac{2 \ln 3}{\varepsilon} \right\rceil$ **times**
12 Sample an index x from $[n]$ uniformly at random and query $A[x]$.
13 **if** $(A[\mathsf{predecessor}(x)], A[x], A[\mathsf{successor}(x)])$ is not in sorted order **then reject**.
14 **Accept**.

▶ **Lemma 3.1.** *For $r < 1/\varepsilon$, Algorithm 2 makes $O\left(\frac{\log r}{\varepsilon}\right)$ queries.*

Proof. We first bound the query complexity of Steps 4-10. Let w be the number of times Steps 4-10 are run by Algorithm 2. For $k \in [0..w]$, let T_k be the snapshot of the binary search tree T of array indices (initialized in Step 3) at the end of iteration k. Note that $T_0 = \{1, n\}$ and T_w is the tree at the end of the algorithm. Let $V_k = \{v : v = A[i] \text{ for some } i \in T_k\}$ be the set of all array values of indices in T_k. Observe that once a value v is in V_k, it remains in $V_{k'}$ for all $k' > k$. For $v \in V_k$, define $\mathsf{successor\text{-}distance}(v, T_k) = |i - \mathsf{successor}(i)|$ such that $A[i] = v$ and $A[\mathsf{successor}(i)] \neq v$, where $\mathsf{successor}$ is defined with respect to the tree T_k (for $v = A[n]$, define $\mathsf{successor\text{-}distance}(v, T_k) = 0$). Consider the k^{th} iteration of Steps 4-10 where $k \in [w]$. In Step 4 of k^{th} iteration, an index $i \in T$ is chosen such that $\mathsf{successor\text{-}distance}(A[i], T_{k-1}) > \varepsilon n/2r$. At the end of the iteration, $\mathsf{successor\text{-}distance}(A[i], T_k) = \mathsf{successor\text{-}distance}(A[i], T_{k-1})/2$ ignoring the errors due to rounding. Generalizing this argument, for each iteration $k \in [w]$, there exists some $v_k \in V_{k-1} \setminus \{A[n]\}$, such that $\mathsf{successor\text{-}distance}(v_k, T_k) = \mathsf{successor\text{-}distance}(v_k, T_{k-1})/2$.

Fix $v^* \in V_w \setminus \{A[n]\}$. Let $k_1, k_2, \ldots, k_q \in [w]$, where $k_1 < k_2 < \ldots < k_q$, be the iterations where the choice of i in Step 4 satisfies $A[i] = v^*$. From the description of the tester, for any $i \in [2..q]$, we have $\mathsf{successor\text{-}distance}(v^*, T_{k_i}) = \mathsf{successor\text{-}distance}(v^*, T_{k_{i-1}})/2$. By extending this relation, we get $\mathsf{successor\text{-}distance}(v^*, T_{k_q}) = \mathsf{successor\text{-}distance}(v^*, T_{k_1})/2^{q-1}$. But $\mathsf{successor\text{-}distance}(v^*, T_{k_1}) < n$ and $\frac{\varepsilon n}{4r} < \mathsf{successor\text{-}distance}(v^*, T_{k_q}) \leq \frac{\varepsilon n}{2r}$. Solving for q, we get

$$2^{q-1} = \frac{\mathsf{successor\text{-}distance}(v^*, T_{k_1})}{\mathsf{successor\text{-}distance}(v^*, T_{k_q})} < \frac{n}{\varepsilon n/4r} = \frac{4r}{\varepsilon};$$

$$q < \log \frac{8r}{\varepsilon}.$$

Hence, the tester runs at most $\log(8r/\varepsilon)$ iterations where $\mathsf{successor\text{-}distance}(v^*, \cdot)$ is halved.

Accounting for all the iterations for each value in $V_w \setminus \{A[n]\}$, we get

$$w < |V_w| \cdot \log(8r/\varepsilon) \leq r \log(8r/\varepsilon),$$

since $|V_w| \leq r$. In each iteration, the tester makes a constant number of queries. So, the overall query complexity of Steps 4-10 is $O\left(r \log \frac{r}{\varepsilon}\right)$. The query complexity of Steps 11-13 is $O(1/\varepsilon)$. Hence, the overall query complexity of the tester is $O\left(\frac{1}{\varepsilon} + r \log \frac{r}{\varepsilon}\right)$.

Now, we prove that $O\left(r \log \frac{r}{\varepsilon}\right) = O\left(\frac{\log r}{\varepsilon}\right)$ for $r < 1/\varepsilon$. We have $O\left(r \log \frac{r}{\varepsilon}\right) = O\left(r \log \frac{1}{\varepsilon}\right)$ as $r < 1/\varepsilon$. Note that the function $g(x) = \frac{x}{\log x}$ is increasing for $x \geq 3$. Hence, for $r < 1/\varepsilon$, we have $\frac{r}{\log r} < \frac{1/\varepsilon}{\log(1/\varepsilon)}$, and hence $r \log \frac{1}{\varepsilon} < \frac{\log r}{\varepsilon}$. Therefore, the query complexity of Algorithm 2 is $O\left(\frac{\log r}{\varepsilon}\right)$. ◀

▶ **Lemma 3.2.** *Steps 2-14 of Algorithm 2, with probability at least 2/3, reject every array that has at most r distinct values and is ε-far from sorted, when $r < 1/\varepsilon$.*

Proof. Consider an array A that has at most r distinct values and is ε-far from sorted, where $r < 1/\varepsilon$. Algorithm 2 rejects whenever it finds elements out of order. We show that Steps 11-13 reject with probability at least 2/3, if Steps 2-10 do not find array elements out of order.

Consider the indices in the tree T at the end of the while loop. Let $E = \{j \in T : A[j] < A[\mathsf{successor}(j)]\}$ be the indices in T whose array values differ from that of their respective successor in T. As A has at most r distinct values, by Pigeonhole principle, $|E| < r$. Each $i \in E$ satisfies $|i - \mathsf{successor}(i)| \leq \varepsilon n/2r$. Define $E' = \{k \in [n] : i < k < \mathsf{successor}(i), i \in E\}$. Clearly, $|E'| \leq \frac{\varepsilon n}{2r} \cdot |E| < \frac{\varepsilon n}{2}$. Consider the subarray of A indexed by $[n] \setminus E'$. This subarray is $\frac{\varepsilon}{2}$-far from sorted as A is ε-far from sorted. Also, all $k \in [n] \setminus E'$ satisfy $\mathsf{predecessor}(k) < k < \mathsf{successor}(k)$ and $A[\mathsf{predecessor}(k)] = A[\mathsf{successor}(k)]$ (note that the definitions of $\mathsf{predecessor}$ and $\mathsf{successor}$ are applicable to all elements in $[n]$). That is, for all such indices k, we know what the element $A[k]$ should be if A is sorted. Recall that if $A[i] = A[j]$, then $[i..j]$ constitutes a *nearly-constant* interval, as defined in Section 2. By the proof method used in Lemma 2.3, there exists at least $\varepsilon n/2$ indices of the form $k \in [n]$ such that $A[\mathsf{predecessor}(k)] = A[\mathsf{successor}(k)]$ and $A[k] \neq A[\mathsf{successor}(k)]$. The probability that Steps 12 and 13 fail to capture such an index in any of its $\left\lceil \frac{2 \ln 3}{\varepsilon} \right\rceil$ iterations is at most

$$(1 - \varepsilon/2)^{\frac{2 \ln 3}{\varepsilon}} \leq 1/3.$$ ◀

4 The Monotonicity Tester over Hypergrids

In this section, we describe a monotonicity tester for functions over hypergrid domains and prove Theorem 1.7. We prove the correctness of this tester using the correctness of the sortedness tester described in Section 3, a dimension reduction theorem by Chakrabarty et al. [15] and the work investment strategy by Berman et al. [9].

An axis-parallel line ℓ of the hypergrid $[n]^d$ is a set of n points that agree on all but one coordinate. Let $f|_\ell$ denote the restriction of a function f to ℓ. Note that $f|_\ell$ can be thought of as a real-valued function over $[n]$.

The tester iteratively samples uniformly random axis-parallel lines, runs Algorithm 2 on each of them, and rejects if any run of Algorithm 2 rejects. We now analyze the tester and prove Theorem 1.7.

Proof of Theorem 1.7. We prove that Algorithm 3 is a 1-sided error ε-tester that makes $O\left(\frac{d}{\varepsilon} \log \frac{d}{\varepsilon} \log r\right)$ queries to test monotonicity of real-valued functions $f : [n]^d \mapsto \mathbb{R}$ over the

Algorithm 3: The Monotonicity Tester over Hypergrids

input: query access to $f : [n]^d \mapsto \mathbb{R}$, an upper bound r on $|\mathrm{Im}(f)|$, and a distance parameter $\varepsilon \in (0, 1)$.

1 **for** $i = 1$ *to* $\left\lceil 3 \log \frac{4d}{\varepsilon} \right\rceil$ **do**
2 **repeat** $\left\lceil \frac{16d \ln 4}{2^i \varepsilon} \right\rceil$ **times**
3 Sample a uniformly random axis-parallel line ℓ.
4 Repeat twice: run Algorithm 2 on the array induced by $f|_\ell$, with the distance parameter set to 2^{-i} and the upper bound on the number of distinct elements set to r; **reject** if it rejects at least once.
5 **Accept**.

hypergrid domain, where $|\mathrm{Im}(f)| \leq r$. Algorithm 3 has 1-sided error because Algorithm 2, which it runs as a subroutine, has 1-sided error. Lemmas 4.1 and 4.2 complete the proof of Theorem 1.7. ◀

▶ **Lemma 4.1.** *Algorithm 3 makes* $O\left(\frac{d}{\varepsilon} \log \frac{d}{\varepsilon} \log r\right)$ *queries.*

Proof. The query complexity of a single execution of Step 4 during the i^{th} iteration of the outermost loop (Step 1) is $O(2^i \log r)$. As Step 4 is repeated $O\left(\frac{d}{2^i \varepsilon}\right)$ times in the i^{th} iteration, the overall query complexity of the i^{th} iteration of the tester is $O\left(\frac{d}{\varepsilon} \log r\right)$. The outermost loop is executed $O(\log \frac{d}{\varepsilon})$ times, and hence the query complexity of the tester is $O\left(\frac{d}{\varepsilon} \log \frac{d}{\varepsilon} \log r\right)$. ◀

▶ **Lemma 4.2.** *Algorithm 3, with probability at least* $2/3$*, rejects every function over the hypergrid domain which is* ε*-far from sorted and has image size is at most* r*.*

Proof. Let $f : [n]^d \mapsto \mathbb{R}$ be ε-far from monotone, with $|Im(f)| \leq r$. Let $\mathcal{L}_{n,d}$ denote the set of all axis-parallel lines in $[n]^d$ and $d_{\mathcal{M}}(f)$ denote the relative distance of f to monotonicity. We also use $d_{\mathcal{M}}(f|_\ell)$ to denote the relative distance to monotonicity of the function $f|_\ell$. We have $|\mathrm{Im}(f|_\ell)| \leq r$ since $|\mathrm{Im}(f)| \leq r$. We use the following dimension reduction theorem proved by Chakrabarty et al. [15].

▶ **Theorem 4.3** (Chakrabarty et al. [15])**.**

$$\mathbb{E}_{\ell \leftarrow \mathcal{L}_{n,d}}[d_{\mathcal{M}}(f|_\ell)] \geq \frac{d_{\mathcal{M}}(f)}{4d}.$$

We note that Theorem 4.3 is a special case of the dimension reduction theorem proved in [15]. Clearly, if $d_{\mathcal{M}}(f) \geq \varepsilon$, then, $\mathbb{E}_{\ell \leftarrow \mathcal{L}_{n,d}}[d_{\mathcal{M}}(f|_\ell)] \geq \varepsilon/4d$. We use the work investment strategy due to Berman et al. [9] to extend the monotonicity tester on the line domain to the hypergrid domain.

▶ **Theorem 4.4** (Berman et al. [9])**.** *For a random variable* $X \in [0, 1]$ *with* $\mathbb{E}[X] \geq \mu$ *for* $\mu < 1/2$*, let* $p_i = \Pr[X \geq \frac{1}{2^i}]$ *and* $\delta \in (0, 1)$ *be the desired probability of error. Let* $k_i = \frac{4 \ln 1/\delta}{2^i \mu}$*. Then,*

$$\prod_{i=1}^{\lceil 3 \log(1/\mu) \rceil} (1 - p_i)^{k_i} \leq \delta.$$

Algorithm 4: The Uniform Sortedness Tester

input: query access to an array A of size n, an upper bound r on the number of distinct values in A, and a distance parameter $\varepsilon \in (0,1)$.

1 Sample $\left\lceil \frac{24\sqrt{r}}{\epsilon} \right\rceil$ indices from A uniformly and independently at random.

2 **Reject** if the array restricted to the sampled indices is not sorted; otherwise, **accept**.

Consider running Algorithm 3 on f. Let $X = d_{\mathcal{M}}(f|_\ell)$, where ℓ is sampled uniformly at random from $\mathcal{L}_{n,d}$. We apply the work investment strategy (Theorem 4.4) on X with error probability $\delta = 1/4$. By Theorem 4.3, $\mathbb{E}[X] \geq \varepsilon/4d$. Thus, in Theorem 4.4, we set $\mu = \varepsilon/4d$ and $k_i = \frac{16d \ln 4}{2^i \varepsilon}$ for all $i \in \left[\left\lceil 3 \log \frac{4d}{\varepsilon} \right\rceil\right]$. By Theorem 4.4, with probability at least 3/4, for some $i \in \left[\left\lceil 3 \log \frac{4d}{\varepsilon} \right\rceil\right]$, we sample a line ℓ such that $d_{\mathcal{M}}(f|_\ell) \geq 2^{-i}$ in Step 3. Conditioned on sampling such a line, Step 4 rejects ℓ with probability at least 8/9. Thus, given a function f that is ε-far from sorted, Algorithm 3 rejects f with probability at least $\frac{3}{4} \cdot \frac{8}{9} = \frac{2}{3}$, as required. This completes the proof of Lemma 4.2. ◀

Note on a nonadaptive tester for hypergrids. We can get a nonadaptive, 1-sided error ε-tester for monotonicity over hypergrids by using Algorithm 1 instead of Algorithm 2 in Step 4 of Algorithm 3. The same analysis goes through for this case and the overall query complexity of the tester is $O\left(\frac{d}{\varepsilon} \log \frac{d}{\varepsilon} \log \frac{rd}{\varepsilon}\right)$.

5 The Uniform Tester for Sortedness

In this section, we describe a nonadaptive ε-tester that makes $O(\sqrt{r}/\varepsilon)$ uniform and independent queries to test sortedness of arrays containing at most r distinct values and prove Theorem 1.5.

Our tester is Algorithm 4. The bound on the query complexity of the tester follows directly from its description. The tester has 1-sided error as it always accepts sorted arrays. In the rest of the section, we show that, with high probability, the tester rejects arrays that are ε-far from sorted.

▶ **Lemma 5.1.** *Algorithm 4, with probability at least 2/3, rejects every array that has at most r distinct elements and is ε-far from sorted.*

Proof. Consider an array A that has at most r distinct values and is ε-far from sorted. Recall that a pair of indices (x, y), where $x, y \in [n]$ and $x < y$, is *violated* in an array A if $A(x) > A(y)$. Consider the undirected *violation graph* $G = ([n], E)$ of A, where an edge $\{u, v\} \in E$ if (u, v) is a violated pair. Dodis et al. [22, Lemma 7] show that if A is ε-far from sorted then G has a matching M of size at least $\varepsilon n/2$.

For a pair $(x, y) \in [n] \times [n]$ such that $x < y$, we refer to x as its lower endpoint and y as its higher endpoint. We first partition the pairs in M into r classes as follows. Let $v_1 < v_2 < \cdots < v_r$ be the values in the range. A pair $(x, y) \in M$ such that $x < y$ belongs to the i^{th} class C_i, if $A(x) = v_i$. Note that C_1 is empty. For each $i \in [r]$, let C_i^{L} and C_i^{H} denote the set of lower and higher endpoints of pairs in C_i, respectively. Note that $|C_i| = |C_i^{\text{L}}| = |C_i^{\text{H}}|$. For each $i \in [r]$, define the i^{th} lower bucket B_i^{L} to consist of the smallest $\lceil |C_i|/2 \rceil$ indices in C_i^{L} and the i^{th} higher bucket B_i^{H} to consist of the largest $\lceil |C_i|/2 \rceil$ indices in C_i^{H}. Note that $\left|\bigcup_{i \in [r]} B_i^{\text{L}}\right| = \left|\bigcup_{i \in [r]} B_i^{\text{H}}\right| \geq \varepsilon n/4$. It is easy to see that for each $i \in [r]$,

every pair in $B_i^{\mathrm{L}} \times B_i^{\mathrm{H}}$ is a violated pair. Therefore, if an algorithm samples indices from both B_i^{L} and B_i^{H}, for some $i \in [r]$, it rejects. To bound the probability of this event from below, we use the following generalization of the Birthday Paradox proved by Goldreich et al. [27, Lemma 19].

▶ **Claim 5.2** ([27, Lemma 19]). Let $S_1, S_2 \ldots, S_r, T_1, T_2 \ldots, T_r$ be disjoint subsets of a universe U. For each $i \in [r]$, let $p_i = |S_i|/|U|$ and $q_i = |T_i|/|U|$. Let $\rho = \sum_i \min\{p_i, q_i\}$. Then, if we uniformly sample $6\sqrt{r}/\rho$ elements from U, with probability at least $2/3$, for some $i \in [r]$, the sample will contain at least one element from both S_i and T_i.

If we set $S_i = B_i^{\mathrm{L}}$ and $T_i = B_i^{\mathrm{H}}$ for each $i \in [r]$ in Claim 5.2, we have $\rho \geq \varepsilon/4$. Therefore, a uniform sample of $24\sqrt{r}/\varepsilon$ points from $[n]$, with probability at least $2/3$, will have, for some $i \in [r]$, an index from B_i^{L} and B_i^{H}, and the algorithm will reject. This completes the proof of the lemma. ◀

6 A Lower Bound for the Uniform Sortedness Tester

In this section, we prove that $\Omega(\sqrt{r})$ uniform queries are required to test sortedness of an array with at most r distinct values, even when one allows for 2-sided error, and prove Theorem 1.6. The proof uses Yao's principle [41], the version with two distributions (see, e.g., Raskhodnikova and Smith [39]). We first define two hard distributions $\mathcal{P}$ and $\widehat{\mathcal{N}}$ on arrays with r distinct values such that every array drawn from $\mathcal{P}$ is in sorted order and every array drawn from $\widehat{\mathcal{N}}$ is $\frac{1}{8}$-far from sorted. We then show that, for any tester that uses $o(\sqrt{r})$ uniform queries, the statistical difference between tester's views of the two distributions is small, and hence, with high probability, it cannot distinguish between the distributions.

The statistical distance between two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$, denoted by $\mathrm{SD}(\mathcal{D}_1, \mathcal{D}_2)$, is defined as

$$\mathrm{SD}(\mathcal{D}_1, \mathcal{D}_2) = \max_{S \subseteq (\mathrm{support}(\mathcal{D}_1) \cup \mathrm{support}(\mathcal{D}_2))} \left(\left| \Pr_{x \leftarrow \mathcal{D}_1}[x \in S] - \Pr_{x \leftarrow \mathcal{D}_2}[x \in S] \right| \right).$$

We write $\mathcal{D}_1 \approx_\delta \mathcal{D}_2$ to denote $\mathrm{SD}(\mathcal{D}_1, \mathcal{D}_2) \leq \delta$.

Proof of Theorem 1.6. First, we define distributions $\mathcal{P}$ and $\mathcal{N}$ on arrays of size n taking values in the set $[r]$, where $n \geq 16r \ln 8r$. Without loss of generality, we assume that r is an even number that divides n.

The distribution $\mathcal{P}$ is constructed as follows. Partition an n-element array into $r/2$ blocks, each of length $2n/r$. For $i \in [r/2]$, set the values in the i^{th} block of the array to $((2i-1), 2i, 2i, \ldots, 2i)$ with probability $1/2$ and $((2i-1), (2i-1), \ldots, (2i-1), 2i)$ with probability $1/2$, independent of the other blocks.

The distribution $\mathcal{N}$ is constructed as follows. As before, partition an n-element array into $r/2$ blocks, each of length $2n/r$. For $i \in [r/2]$, set the first value in the i^{th} block to $(2i-1)$ and the last value to $2i$. The values at all other indices in that block are set to either $(2i-1)$ or $2i$ uniformly and independently at random.

Note that every array drawn from $\mathcal{P}$ is in sorted order. We will show that, with high probability, an array drawn from $\mathcal{N}$ is $\frac{1}{8}$-far from sorted.

▶ **Lemma 6.1.** *Let E denote the event that an array chosen according to $\mathcal{N}$ is $\frac{1}{8}$-far from sorted. Then,*

$$\Pr[E] > \frac{5}{6}.$$

Proof. Consider an array A chosen according to $\mathcal{N}$. Consider the i^{th} block of A for some $i \in [r/2]$. Let Y_{2i} denote the number of elements with value $2i$ in the first half of this block and Y_{2i-1} denote the number of elements with value $(2i-1)$ in the second half of the block. As the size of each half of the block is n/r, and the value at each index (except for the first and the last index) is assigned either $(2i-1)$ or $2i$ uniformly and independently at random,

$$\mathbb{E}[Y_{2i}] = \mathbb{E}[Y_{2i-1}] = \frac{n}{2r} - \frac{1}{2}.$$

By a Chernoff bound, for all $i \in \left[\frac{r}{2}\right]$,

$$\begin{aligned}\Pr\left[Y_{2i} \le \frac{n}{4r}\right] = \Pr\left[Y_{2i-1} \le \frac{n}{4r}\right] &= \Pr\left[Y_{2i-1} \le \left(1 - \frac{n-2r}{2(n-r)}\right)\left(\frac{n}{2r} - \frac{1}{2}\right)\right] \\ &\le \exp\left(-\frac{n}{16r} \cdot \frac{(n-2r)^2}{n(n-r)}\right) \\ &< \frac{1}{6r}.\end{aligned}$$

If $Y_{2i} > n/4r$ and $Y_{2i-1} > n/4r$, then at least $n/4r$ elements in i^{th} block need to be changed to make it sorted, as all the indices with value $2i$ in the first half or all the indices with value $2i-1$ in the last half need to be changed. By the union bound,

$$\Pr\left[\bigvee_{j=1}^{r}\left(Y_j \le \frac{n}{4r}\right)\right] \le r \cdot \Pr\left[Y_1 \le \frac{n}{4r}\right] < \frac{1}{6}.$$

With probability at least $5/6$, we have $Y_{2i} > n/4r$ and $Y_{2i-1} > n/4r$ for all $i \in [r/2]$. This implies that at least $n/4r$ elements need to be changed in each of the $r/2$ blocks to make it sorted. Hence, with probability at least $5/6$, the array A is $\frac{1}{8}$-far from sorted. ◀

Denote the conditional distribution $\mathcal{N}|_E$ by $\widehat{\mathcal{N}}$. Any instance sampled according to $\widehat{\mathcal{N}}$ is $\frac{1}{8}$-far from sorted. The statistical distance $\mathrm{SD}(\mathcal{N}, \widehat{\mathcal{N}})$ can be bounded using the following lemma proven by Raskhodnikova and Smith [39].

▶ **Lemma 6.2** ([39, Claim 4]). *Let E be an event that happens with probability at least $1-\delta$ under the distribution $\mathcal{D}$. Then, $\mathcal{D} \approx_{\delta'} \mathcal{D}|_E$, where $\delta' = \frac{1}{1-\delta} - 1$.*

Applying Lemma 6.2 to $\mathcal{N}$ and $\widehat{\mathcal{N}}$, we get $\mathcal{N} \approx_{1/5} \widehat{\mathcal{N}}$.

Consider any $\frac{1}{8}$-tester for sortedness that makes q queries where $q \le \sqrt{r}/5$. Define $\mathcal{P}$-view to be the distribution of values at the q locations queried by the tester in an array sampled according to $\mathcal{P}$. Similarly, define $\mathcal{N}$-view and $\widehat{\mathcal{N}}$-view. Next, we show that it is hard to distinguish $\mathcal{P}$-view from $\widehat{\mathcal{N}}$-view.

▶ **Lemma 6.3.**

$$SD(\mathcal{P}\text{-view}, \widehat{\mathcal{N}}\text{-view}) < \frac{1}{3}.$$

Proof. Let F denote the event that at least 2 out of the tester's q uniform samples from an array A are from the same block. An upper bound on the probability of this event can be obtained using the following lemma.

▶ **Lemma 6.4** (Bellare and Rogaway [3]). *Consider q balls and N bins, where each ball is assigned uniformly and independently at random to one of the bins. The probability that there exists a pair of balls assigned to the same bin is at most $\frac{q(q-1)}{2N}$.*

Algorithm 5: The Convexity Tester

input: query access to $f : [n] \mapsto \mathbb{R}$, an upper bound r on $|\mathrm{Im}(f)|$, and a distance parameter $\varepsilon \in (0, 1)$.

if $r \geq \frac{\varepsilon n}{3}$ **then**

1 Run the ε-tester for convexity by Parnas et al. [37] on f and **reject** if it rejects.

else

2 Let $M \leftarrow [r + 1, \ldots, n - r]$.

3 Sample $\lceil \frac{4}{\varepsilon} \rceil$ indices from M uniformly and independently at random.

4 **Reject** if f restricted to those indices is not constant.

5 **Accept**.

By Lemma 6.4, we get $\Pr[F] \leq \frac{q(q-1)}{2 \cdot r/2} < \frac{q^2}{r} = \frac{1}{25}$. Then, by Lemma 6.2,

$$\mathcal{P}\text{-view} \approx_{1/24} \mathcal{P}\text{-view}|_{\overline{F}}; \tag{1}$$

$$\mathcal{N}\text{-view} \approx_{1/24} \mathcal{N}\text{-view}|_{\overline{F}}. \tag{2}$$

Since $\mathcal{N} \approx_{1/5} \widehat{\mathcal{N}}$, the definition of statistical difference implies that

$$\mathcal{N}\text{-view} \approx_{1/5} \widehat{\mathcal{N}}\text{-view}. \tag{3}$$

It remains to show that $\mathcal{P}\text{-view}|_{\overline{F}} = \mathcal{N}\text{-view}|_{\overline{F}}$. Let x be an index in the i^{th} block, for some $i \in [r/2]$. If x is neither the first nor the last index of i^{th} block, then $\Pr[A[x] = (2i-1)] = \Pr[A[x] = 2i] = 1/2$ irrespective of whether $A \leftarrow \mathcal{P}$ or $A \leftarrow \mathcal{N}$. If x is the first or the last index of the i^{th} block, then $A[x]$ is fixed to the same value under both $\mathcal{P}$ and $\mathcal{N}$. If $\overline{F}$ holds, then at most 1 index from each block is sampled by the tester. By the definition of $\mathcal{P}$ and $\mathcal{N}$, for any two indices from different blocks, the values assigned to them are independent of each other. Hence, $\mathcal{P}\text{-view}|_{\overline{F}} = \mathcal{N}\text{-view}|_{\overline{F}}$. By (1)-(3),

$$\mathrm{SD}(\mathcal{P}\text{-view}, \widehat{\mathcal{N}}\text{-view}) \leq \frac{1}{24} + \frac{1}{24} + \frac{1}{5} < \frac{1}{3}.$$

This completes the proof of Lemma 6.3. ◀

By Yao's principle [41], as stated in [39, Claim 5], for $q \leq \sqrt{r}/5$, it is hard for any $\frac{1}{8}$-tester using q uniform queries to distinguish $\mathcal{P}$ from $\widehat{\mathcal{N}}$. Thus, uniform testers for sortedness of arrays with values in $[r]$ require $\Omega(\sqrt{r})$ queries. This completes the proof of Theorem 1.6. ◀

7 Testing Convexity

In this section, we describe a nonadaptive tester for convexity of functions $f : [n] \mapsto \mathbb{R}$ and prove Theorem 1.8. Recall that a function $f : [n] \mapsto \mathbb{R}$ is convex if $f(i)-f(i-1) \leq f(i+1)-f(i)$ for $1 < i \leq n$. Our convexity tester is Algorithm 5. It uses the nonadaptive convexity tester of Parnas et al. [37] as a black box.

The query complexity of our tester is $O(1/\varepsilon)$ when $r < \varepsilon n/3$, as is evident from its description. In the other case, $n \leq 3r/\varepsilon$, our tester runs the tester of [37], which makes $O(\log n/\varepsilon)$ queries. Substituting the upper bound on n, we get the query complexity bound claimed in Theorem 1.8.

Given a function $f : [n] \mapsto \mathbb{R}$ and a set $S \subseteq [n]$, let f_S denote the restriction of f to the indices in S whenever $S \neq \emptyset$. To prove the correctness of our tester, we first prove the following characterization of convex functions with image size at most r.

▶ **Claim 7.1.** If $f : [n] \mapsto \mathbb{R}$ is convex and $|\text{Im}(f)| \leq r$, then f_M for $M = [r+1..n-r]$ is a constant function.

Proof. We can assume that $r < n/2$, for otherwise, $M = \emptyset$. Assume for the sake of contradiction that there exists points $x, x+1 \in M$ such that $f_M(x) \neq f_M(x+1)$. If $f_M(x) < f_M(x+1)$, then f has to be monotonically increasing on the domain restricted to $[x+1, \ldots, n]$, which has more than r elements in it as $x < n - r + 1$. By the pigeonhole principle, this results in a contradiction, as $|\text{Im}(f)| \leq r$. If $f_M(x) > f_M(x+1)$, then f has to be monotonically decreasing on the set $[1, \ldots, x+1]$, which has more than r elements in it since $x \geq r$. By the pigeonhole principle, this also leads to a contradiction, as $|\text{Im}(f)| \leq r$. Hence, f can take only one value on M and therefore, f_M is a constant function. ◀

We will now show that the tester accepts every function that is convex and rejects with probability at least 2/3, every function that is ε-far from convex.

▶ **Lemma 7.2.** *Consider a function $f : [n] \mapsto \mathbb{R}$. Algorithm 5, on input $r \geq |Im(f)|$ and ε, accepts if f is convex and rejects, with probability at least 2/3, if f is ε-far from convex.*

Proof. If $r \geq \frac{\varepsilon n}{3}$, Algorithm 5 runs the tester for convexity by [37], and so the correctness follows from their analysis.

Consider the case where $r < \varepsilon n/3$. It follows from Claim 7.1 that Algorithm 5 accepts f if it is convex. Now assume that f is ε-far from convex. It remains to prove that f_M is $\varepsilon/3$-far from being a constant function, where $M = [r+1, \ldots, n-r]$. Assume for the sake of contradiction that f_M is $\varepsilon/3$-close to constant. We will construct a convex function $g : [n] \mapsto \mathbb{R}$ such that g is ε-close to f and satisfies $|\text{Im}(g)| \leq r$. This will give us the required contradiction. Let the constant function closest to f_M be h, where $h(x) = c$ for every $x \in M$. The function g is then defined as a constant function taking the value c on all points in $[n]$. Since the Hamming distance of f_M from h is at most $\varepsilon n/3$, the total Hamming distance of f from g is at most $\varepsilon n/3 + 2r < \varepsilon n$. This contradicts the fact that f is ε-far from convex. Hence, f_M is $\frac{\varepsilon}{3}$-far from being a constant function. The probability that $4/\varepsilon$ samples fail to detect that f_M is $\varepsilon/3$-far from constant is at most $(1 - \varepsilon/3)^{4/\varepsilon} \leq \exp(-4/3) < 1/3$. ◀

References

1. Nir Ailon and Bernard Chazelle. Information theory in property testing and monotonicity testing in higher dimension. *Inf. Comput.*, 204(11):1704–1717, 2006.
2. Tugkan Batu, Ronitt Rubinfeld, and Patrick White. Fast approximate PCPs for multidimensional bin-packing problems. *Inf. Comput.*, 196(1):42–56, 2005.
3. Mihir Bellare and Phillip Rogaway. Lecture notes on modern cryptography, 2005. URL: `https://cseweb.ucsd.edu/~mihir/cse207/w-birthday.pdf`.
4. Aleksandrs Belovs and Eric Blais. Quantum algorithm for monotonicity testing on the hypercube. *Theory of Computing*, 11:403–412, 2015.
5. Aleksandrs Belovs and Eric Blais. A polynomial lower bound for testing monotonicity. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 1021–1032, 2016.
6. Sagi Ben-Moshe, Yaron Kanza, Eldar Fischer, Arie Matsliah, Mani Fischer, and Carl Staelin. Detecting and exploiting near-sortedness for efficient relational query evaluation. In *Database Theory - ICDT 2011, 14th International Conference, Uppsala, Sweden, March 21-24, 2011, Proceedings*, pages 256–267, 2011.
7. Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. The power and limitations of uniform samples in testing properties of figures. In *36th IARCS Annual Conference*

on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2016, December 13-15, 2016, Chennai, India, pages 45:1–45:14, 2016.

8 Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. Testing convexity of figures under the uniform distribution. In *32nd International Symposium on Computational Geometry, SoCG 2016, June 14-18, 2016, Boston, MA, USA*, pages 17:1–17:15, 2016.

9 Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. L_p-testing. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 164–173, 2014.

10 Arnab Bhattacharyya, Elena Grigorescu, Kyomin Jung, Sofya Raskhodnikova, and David P. Woodruff. Transitive-closure spanners. *SIAM J. Comput.*, 41(6):1380–1425, 2012.

11 Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. *Computational Complexity*, 21(2):311–358, 2012.

12 Eric Blais, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Lower bounds for testing properties of functions over hypergrid domains. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 309–320, 2014.

13 Jop Briët, Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Monotonicity testing and shortest-path routing on the cube. *Combinatorica*, 32(1):35–53, 2012.

14 Deeparnab Chakrabarty. Monotonicity testing. In *Encyclopedia of Algorithms*, pages 1352–1356. Springer, 2016.

15 Deeparnab Chakrabarty, Kashyap Dixit, Madhav Jha, and C. Seshadhri. Property testing on product distributions: Optimal testers for bounded derivative properties. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1809–1828, 2015.

16 Deeparnab Chakrabarty and C. Seshadhri. Optimal bounds for monotonicity and Lipschitz testing over hypercubes and hypergrids. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 419–428, 2013.

17 Deeparnab Chakrabarty and C. Seshadhri. An optimal lower bound for monotonicity testing over hypergrids. *Theory of Computing*, 10:453–464, 2014.

18 Deeparnab Chakrabarty and C. Seshadhri. An $o(n)$ monotonicity tester for Boolean functions over the hypercube. *SIAM J. Comput.*, 45(2):461–472, 2016.

19 Xi Chen, Anindya De, Rocco A. Servedio, and Li-Yang Tan. Boolean function monotonicity testing requires (almost) $n^{1/2}$ non-adaptive queries. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, STOC'15, pages 519–528, New York, NY, USA, 2015.

20 Xi Chen, Rocco A. Servedio, and Li-Yang Tan. New algorithms and lower bounds for monotonicity testing. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 286–295, 2014.

21 Kashyap Dixit, Sofya Raskhodnikova, Abhradeep Thakurta, and Nithin M. Varma. Erasure-resilient property testing. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 91:1–91:15, 2016.

22 Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. In *RANDOM-APPROX'99, Berkeley, CA, USA, August 8-11, 1999, Proceedings*, pages 97–108, 1999.

23 Funda Ergün, Sampath Kannan, Ravi Kumar, Ronitt Rubinfeld, and Mahesh Viswanathan. Spot-checkers. *J. Comput. Syst. Sci.*, 60(3):717–751, 2000.

24 Eldar Fischer. On the strength of comparisons in property testing. *Inf. Comput.*, 189(1):107–116, 2004.

25 Eldar Fischer, Oded Lachish, and Yadu Vasudev. Trading query complexity for sample-based testing and multi-testing scalability. In *IEEE 56th Annual Symposium on Founda-*

tions of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015, pages 1163–1182, 2015.

26 Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 474–483, 2002.

27 Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samorodnitsky. Testing monotonicity. *Combinatorica*, 20(3):301–337, 2000.

28 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998.

29 Oded Goldreich and Dana Ron. On proximity-oblivious testing. *SIAM J. Comput.*, 40(2):534–566, 2011.

30 Oded Goldreich and Dana Ron. On sample-based testers. *TOCT*, 8(2):7, 2016.

31 Shirley Halevy and Eyal Kushilevitz. A lower bound for distribution-free monotonicity testing. In *APPROX-RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings*, pages 330–341, 2005.

32 Shirley Halevy and Eyal Kushilevitz. Testing monotonicity over graph products. *Random Struct. Algorithms*, 33(1):44–67, 2008.

33 Kazuo Iwama and Yuichi Yoshida. Parameterized testability. In *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 507–516, 2014.

34 Madhav Jha and Sofya Raskhodnikova. Testing and reconstruction of Lipschitz functions with applications to data privacy. *SIAM J. Comput.*, 42(2):700–731, 2013.

35 Subhash Khot, Dor Minzer, and Muli Safra. On monotonicity testing and Boolean isoperimetric type theorems. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 52–58, 2015.

36 Eric Lehman and Dana Ron. On disjoint chains of subsets. *J. Comb. Theory, Ser. A*, 94(2):399–404, 2001.

37 Michal Parnas, Dana Ron, and Ronitt Rubinfeld. On testing convexity and submodularity. *SIAM J. Comput.*, 32(5):1158–1184, 2003.

38 Sofya Raskhodnikova. Testing if an array is sorted. In *Encyclopedia of Algorithms*, pages 2219–2222. Springer, 2016.

39 Sofya Raskhodnikova and Adam D. Smith. A note on adaptivity in testing properties of bounded degree graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 13(089), 2006.

40 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.

41 Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity (extended abstract). In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 222–227, 1977.

The Complexity of Problems in P Given Correlated Instances*

Shafi Goldwasser[1] and Dhiraj Holden[2]

1 **Massachusetts Institute of Technology, Cambridge, USA**
shafi@theory.csail.mit.edu
2 **Massachusetts Institute of Technology, Cambridge, USA**
dholden@mit.edu

Abstract

Instances of computational problems do not exist in isolation. Rather, multiple and correlated instances of the same problem arise naturally in the real world. The challenge is how to gain computationally from correlations when they can be found. [DGH, ITCS 2015] showed that significant computational gains can be made by having access to auxiliary instances which are correlated to the primary problem instance *via the solution space*. They demonstrate this for constraint satisfaction problems, which are *NP-hard* in the general worst case form.

Here, we set out to study the impact of having access to correlated instances on the complexity of *polynomial time problems*. Namely, for a problem P that is conjectured to require time n^c for $c > 0$, we ask whether access to a few instances of P that are correlated in some natural way can be used to solve P on one of them (the designated "primary instance") faster than the conjectured lower bound of n^c.

We focus our attention on a number of problems: the Longest Common Subsequence (LCS), the minimum Edit Distance between sequences, and Dynamic Time Warping Distance (DTWD) of curves, for all of which the best known algorithms achieve $\tilde{O}(n^2)$ runtime via dynamic programming. These problems form an interesting case in point to study, as it has been shown that a $O(n^{2-\epsilon})$ time algorithm for a worst-case instance would imply improved algorithms for a host of other problems as well as disprove complexity hypotheses such as the *Strong Exponential Time Hypothesis*.

We show how to use access to a logarithmic number of auxiliary correlated instances, to design novel $o(n^2)$ time algorithms for LCS, EDIT, DTWD, and more generally improved algorithms for *computing any tuple-based similarity measure* - a generalization which we define within on strings. For the multiple sequence alignment problem on k strings, this yields an $O(nk \log n)$ algorithm contrasting with classical $O(n^k)$ dynamic programming.

Our results hold for several correlation models between the primary and the auxiliary instances. In the most general correlation model we address, we assume that the primary instance is a worst-case instance and the auxiliary instances are chosen with uniform distribution subject to the constraint that their allignments are ϵ-correlated with the optimal allignment of the primary instance. We emphasize that optimal solutions for the auxiliary instances will not generally coincide withan optimal solutions for the worst case primary instance.

We view our work as pointing out a new avenue for looking for significant improvements for sequence alignment problems and computing similarity measures, by taking advantage of access to sequences which are correlated through natural generating processes. In this first work we show how to take advantage of mathematically inspired simple clean models of correlation – the intriguing question, looking forward, is to find correlation models which coincide with evolutionary models and other relationships and for which our approach to multiple sequence alignment gives provable guarantees.

* This work was supported by an Akamai Presidential Fellowship, NSF MACS - CNS-1413920, and SIMONS Investigator award Agreement Dated 6-5-12

© Shafi Goldwasser and Dhiraj Holden;
licensed under Creative Commons License CC-BY

8th Innovations in Theoretical Computer Science Conference (ITCS 2017).
Editor: Christos H. Papadimitrou; Article No. 13; pp. 13:1–13:19

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Correlated instances, Longest Common Subsequence, Fine-grained complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.13

1 Introduction

A recent work of Dinur, Goldwasser, and Lin [6] which appeared in ITCS 2015 puts forth the following thesis: "Instances of computational problems of interest rarely exist in isolation, rather, multiple and correlated instances of the same problem arise naturally in the real world. The challenge is not to find settings which present correlated instances, but how to take advantage of correlations when they exist." Obvious examples are in biology where learning genotype to phenotype mappings of an organism can draw on data from multiple specimens with highly correlated DNA; or when multiple files are stored on (or transmitted over) the same error-prone medium, then error patterns in one file may be correlated to error patterns in other files which can be useful for error detection and recovery.

Indeed [DGH, ITCS 2015] show that significant computational gains can be made in solving instances of intractable combinatorial problems, when additional auxiliary instances are available which are *correlated to the primary instance via the solution space*. In particular, they show for several correlation models on the auxiliary instances polynomial time algorithms for constraint satisfaction problems which are NP-hard in the traditional worst case single-instance setting. Similarly, the work of Heninger et al [8] provides examples of number-theoretic problems where correlated cryptographic keys for the RSA function chosen due to poorly designed pseudorandom number generators enable efficient factorization of the RSA moduli involved. The works of [6] and [8] show that access to a polynomial number of auxiliary (but correlated) instances can enable us to find a polynomial time solution to an otherwise **intractable problem**.

In this paper, we set out to study the impact of having access to correlated instances on the complexity of **tractable problems**. Namely, for a polynomial time problem P, we ask whether we can use access to multiple instances of P that are correlated in some natural way, to solve P on one of them – the designated "primary instance" of size n– faster than the conjectured lower bound of n^c.

The question of which polynomial time problems to study is non-obvious. To draw meaningful conclusions from an algorithmic improvement to a polynomial time problem, one should choose problems which are at the same time classical and well-studied and for which there is hopefully some formal evidence to the complexity barrier they face, possibly in the form of a conditional lower bound. An approach for showing conditional lower bounds on a problem P which has gained significant prominence in the last few years and dates back to 1995[7], is to assume a lower bound on the complexity of one long standing open problem Q such as 3SUM or orthogonal-vectors and show a fine-grained reduction of Q to P which implies that any significant algorithmic improvement to P will imply an algorithmic improvement to Q. Another approach has been to show exact polynomial time lower bounds on P under complexity conjectures such as the Strong Exponential Time Hypothesis(SETH) which was introduced by Impagliazzo and Paturi[10] and asserts that solving satisfiability cannot be done significantly faster than exhaustive search.

This "conditional lower bound approach" has been successfully applied, as we detail below, [4, 1, 5] to various problems which compute similarity measures between strings, sequences

and curves, including computing the Longest Common Subsequence (LCS) between pairs and k-tuples of strings (kLCS), computing the minimal Edit Distance (EDIT) between pairs of strings and the dynamic time warping distance (DTWD) between curves. All these problems are currently addressed by dynamic programming algorithms which run in *quadratic time*.

These particular problems are not only the focus of intense theoretical study in the last few years, but arise in natural real-world settings according to some natural generating processes which is suggestive of the availability of multiple correlated instances. For example, the LCS and EDIT sequence alignment problems are often used in identifying conserved sequence regions across a group of sequences of DNA, RNA or proteins hypothesized to be evolutionarily related, or as aid in establishing evolutionary relationships by constructing phylogenetic trees. The dynamic time warping distance applied to signals over time can compare temporal data such as video and audio to detect whether an audio or video sequence is a sped up version of another audio or video sequence. Finally, we remark that to the best of our knowledge there are no known sub-quadratic algorithms for solving LCS, EDIT or DTWD on uniform (vs. worst case) input distributions nor are conditional lower bounds known, although this fascinating challenge has been raised in a multitude of works.

Looking ahead, in this work we will study and demonstrate how the availability of a logarithmic number of auxiliary instances which are correlated to primary instances of the LCS, EDIT and DTWD problems for several natural correlation modes enables us to overcome the quadratic complexity barrier.

But which correlation model over auxiliary instances should be assumed? In this first work we show how to take advantage of mathematically inspired simple and yet natural and clean models of correlation. The intriguing question, looking forward, is to find correlation models which coincide with evolutionary models and other relationships and for which our approach to multiple sequence alignment gives provable guarantees. We view our work as pointing out a novel direction to look for significant improvements, by taking advantage of access to sequences which are correlated through natural processes.

Our algorithmic approach departs from the dynamic programming approaches. It builds on the primary and auxiliary instances to construct new instances over a slightly larger alphabet for which solving the LCS (EDIT and DTWD respectively) can be done in sub-quadratic time and from which the optimal answer for the primary instance can be extracted. Rather than develop a different algorithm for each problem, we define a general notion of a tuple-based similarity measure which captures the LCS, EDIT and DTWD similarity measures on strings as special cases. We will show an algorithm for computing any tuple-based similarity measure over strings as long as access to correlated instances is available.

2 The New Results

We set out to study the impact of access to correlated instances on the complexity of polynomial time problems. The immediate questions which comes to mind is which problems and for which correlations

2.1 The problems: Any tuple-based Similarity Measure

We address the following problems.

- *Longest Common Subsequence* (LCS): Given two strings x and y over an alphabet Σ, find the maximum length of not necessarily consecutive substring common to both.
- *Minimum Edit Distance* (EDIT): Given two strings x and y over alphabet Σ, find the minimum number of insert character, delete character and replace character by another

character operations to apply to x to obtain y.

- *Dynamic Time Warping Distance (DTWD)*: Given two input strings x and y over an alphabet Σ compute the minimal cost of any allowed traversal of the two strings from first character to last where at each traversal step, one advances forward on at least x or y, and the cost is the sum over all time steps of the distance between the current entries. Namely, the sum of the distances between each pair of the traversal When the distances are restricted to $\{0,1\}$ we refer to the problem as $DTWD_{0,1}$
- *k-Longest Common Subsequence* (k-LCS) : Given k strings over alphabet Σ, find the maximum length not necessarily consecutive substring common to all k strings

Each of the above problems computes a different measure of similarity between pairs (or tuples) of strings, but all share a common feature which makes them amenable to speedups using correlated instances. Namely, the value of each of these similarity measures is the optimum over a set of values which only depend on the locations of the alignment. When this is the case, we will show that we can use a correlation model where the alignment stays the same (or with higher probability than uniform stays the same) in the correlated instances, to give a faster algorithm for finding this optimal alignment.

Formally, we define *tuple based similarity measure* which generalizes of all of the above and design an algorithm for any tuple based similarity measure, instead of treating each problem separately.

Main Definition: Let Σ denote an alphabet and let $f : \Sigma^{m_1} \times \Sigma^{m_2} \times, ..., \times \Sigma^{m_k} \to \mathbb{R}$ be a real valued function over k strings of lengths $m_1, ..., m_k$ whose output we informally think of as a measure of similarity between the strings. We say that f is a **tuple-based similarity measure** if there exists:

- a set of allowable k-tuples $\mathcal{S} \subseteq \bigcup_{m_1,m_2,...,m_k} \mathcal{P}([m_1] \times [m_2] \times ... \times [m_k])$ where $\mathcal{P}$ denotes the power set, (these can be viewed as the set of allowable "pseudo-alignments" where multiple positions in one string can match to the same position in another string) and
- a cost function $\mathcal{C}$ defined on pseudo-alignments $S \in \mathcal{S}$ s.t. $\mathcal{C}(S)$ is computable in time $\tilde{O}(|S|)$ and $f(x_1, x_2, ..., x_k) = \max\{\mathcal{C}(S)\text{: } S \in \mathcal{S} \text{ s.t. } \forall (i_1, i_2, ..., i_k) \in S, x_1[i_1] = x_2[i_2] = ... = x_k[i_k]\}$
  (or $f(x_1, x_2, ..., x_k) = \min\{\mathcal{C}(S)\text{: } S \in \mathcal{S} \text{ s.t. } \forall (i_1, i_2, ..., i_k) \in S, x_1[i_1] = x_2[i_2] = ... = x_k[i_k]\}$)

We note that this definition provides a generalization of the alignment gadget of [5] (where $k = 2$) which captures what is needed to have an improved algorithm given correlated instances. Bringmann and Kunnemann [5] introduce the notion of an alignment gadget, and show that any problem with an alignment gadget cannot be solved in strongly subquadratic time. Our definition is more general than theirs in that we do not restrict ourselves to looking at alignments: multiple positions in one string can match to the same position in another string. On the other hand, we only look at measures where the function is determined by the positions of the matches, whereas the gadgets of [5] speak of general measures.

It is easy to see that LCS is a pair-based similarity measure when we take $\mathcal{C}(S) = |S|$; the allowed sets of tuples are all possible alignments, and the maximum over all sets of alignments of $\mathcal{C}(S)$ is precisely the length of the longest common subsequence. In section 3, we prove that computing EDIT and DTWD are each a special case of a pair-based similarity measure.

2.2 The Models of Correlation Models over Auxiliary: Exact and Relaxed

The choices of which auxiliary instances (and which correlations) to consider for the LCS, k-LCS, $EDIT$, and $DTWD$ problems are tailor-made to the problems themselves. Indeed, one may argue that this will likely always be the case, as our hope is to consider correlations which would come up via *natural* generating processes of problem instances.

For example, in bioinformatics, sequence alignment is a way of arranging the sequences of DNA, RNA, or proteins to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences. Although we expect that only certain parts of the genome will be in common between multiple organisms, these parts will have a much higher rate of matching than random chance. Thus, which correlations between genomic sequences one may expect are dictated by evolutionary process which would be highly problem specific.

To describe distributions over the auxiliary instances, we use the framework of generating processes suggested by [6] to describe distributions over instances where correlations are defined between the *solutions* of the primary and auxiliary instance. A generating process $G_{Pri,Aux,t}$ for a problem P proceeds as follows: initialize the process with primary instance I of P drawn from primary distribution Pri. Let S be an underlying solution to I. Then choose t auxiliary instances $\{I_j\}_{j=1,...,t}$ from a distribution Aux conditioned on S in some fashion. The algorithm designer is given the tuple of $t+1$ instances $I, I_1, ..., I_t$ and is tasked with finding a solution to I.

We consider two generating models, which apply for any tuple-based similarity measure, but are easiest to first illustrate for the LCS problem.

Let Pri denote a distribution over pairs of strings of lengths m over alphabet Σ.

Exact LCS Correlation Model GenLCS$_{Pri}(m,t)$: Draw a pair of strings (x,y) from Pri. Let the longest common sub-sequence of (x,y) be at locations $\vec{a} = (a_1, ...a_n)$ in x and locations $\vec{b} = (b_1, ..., b_n)$ in y. Then choose auxiliary instances (x^1, y^1),...,(x^t, x^t) uniformly in $\Sigma^{m_1} \times \Sigma^{m_2}$ subject to the condition that each pair (x^j, y^j) contains a common sub-sequence at locations $\vec{a}$ and $\vec{b}$. Namely, $x^j_{a_i} = y^j_{b_i}$ for all j, for $1 \leq i \leq n$.

We next relax the restriction that each of the auxiliary instance pairs (x^j, y^j) contain a common (although not the longest) subsequence at the locations $\vec{a}$ and $\vec{b}$ of the longest common subsequence of the primary instance (x,y). Instead, we require that in each of the locations in (x^j, y^j) which coincide with locations of the longest common subsequence of x and y, the strings x^j and y^j agree with probability slightly higher than $\frac{1}{2}$.

Let Pri denote a distribution over pairs of strings of lengths m over alphabet Σ.

Relaxed LCS Correlation Model GenLCS2$_{Pri}(m,t,\epsilon)$: Draw a pair of strings (x,y) of length m each from Pri. Let the longest common sub-sequence of (x,y) be at locations $\vec{a} = (a_1, ...a_n)$ in x and locations $\vec{b} = (b_1, ..., b_n)$ in y. Then choose auxiliary instances (x^1, y^1),...,(x^t, x^t) uniformly in $\Sigma^m \times \Sigma^m$. Next, for each pair (x^j, y^j) and for each $1 \leq i \leq n$,set $x^j[a_i] := y^j[b_i]$ with probability ϵ and leave unchanged with probability $1-\epsilon$.

Correlations Models for General Tuple Based Similarity Measures

We are now ready to describe the models stated in terms of *any tuple-based similarity measure on strings*. We again consider two generating processes, an exact and a relaxed one.

The first generating process draws primary instances from a primary distribution, and auxiliary instances from the auxiliary distribution conditioned on the optimal set of tuples for the primary instances matching in each auxiliary instance. The second generating process will be similar to the first generating process, but instead of making the contents of the locations indicated by the optimum tuple always match in the auxiliary instances, there will only be a higher probability of matching in these locations.

Let f be a tuple-based similarity measure with allowed sets of tuples $\mathcal{S}$ and cost function $\mathcal{C}$.

Exact General Generating Model $Gen_{Pri,Aux}(f,k,\vec{m},t)$**:** First draw from the primary distribution Pri strings $x_1,x_2,...,x_k$ of lengths $\vec{m}=m_1,m_2,...,m_k$ where location tuples $S=\{(a_{11},a_{21},...,a_{k1}),...,(a_{1n},a_{2m},...,a_{kn})\}=\arg\min\{\mathcal{C}(S):S\in\mathcal{S}$ s.t. $\forall(i_1,i_2,...,i_k)\in S,x_1[i_1]=x_2[i_2]=...=x_k[i_k]\}$. Next, draw t auxiliary tuples $(x_1^{(i)},x_2^{(i)},...,x_k^{(i)})$ from distribution Aux conditioned on $\forall(j_1,j_2,...,j_k)\in S,x_1^{(i)}[j_1]=x_2^{(i)}[j_2]=...:=x_k^{(i)}[j_k]$ $\forall 1\leq i\leq t$ and output the primary and auxiliary instances drawn.

The main restriction in the above correlation model is that the optimal set of tuple locations must also induce a matching set of tuples for every auxiliary instance. Let us now relax this restriction to give faster algorithms when the optimal set of tuples only has a higher probability of inducing a matching set of tuples in the auxiliary instances.

Relaxed Generating Model $Gen_{Pri,Aux}(f,k,\vec{m},t,\epsilon)$**:** First draw from the primary distribution Pri strings $x_1,x_2,...,x_k$ of lengths
$\vec{m}=m_1,m_2,...,m_k$ where location tuples $S=\{(a_{11},a_{21},...,a_{k1}),...,(a_{1n},a_{2m},...,a_{kn})\}=\arg\min\{\mathcal{C}(S):S\in\mathcal{S}$ s.t. $\forall(i_1,i_2,...,i_k)\in S,x_1[i_1]=x_2[i_2]=...=x_k[i_k]\}$. Next, draws t auxiliary tuples $(x_1^{(i)},x_2^{(i)},...,x_k^{(i)})$ from distribution Aux Finally, set $x_1^{(i)}[j_1]=x_2^{(i)}[j_2]=...:=x_k^{(i)}[j_k]$ with probability ϵ for all tuples $(j_1,j_2,...,j_k)\in S$ and output the primary and auxiliary instances drawn.

Important Discussion on on Correlation Models: A word of caution is in order. One should be careful not to consider correlations which trivialize the problem.

- An example of such trivializations would be access to "too many" auxiliary instances. Whereas in [6] "too many" would correspond to more than a polynomial number of instances, in the current work it would be more than a linear number of auxiliary instances. Indeed, a logarithmic number of additional instances suffice to get a marked improvement.
- Another forbidden trivialization would be access to auxiliary "easy" instances for which the correlation enables trivial extraction of the answer for the primary instance. This is avoided in the exact correlation model as follows. Recall that the primary instance (x,y) can be selected in a worst case manner, whereas the auxiliary instances are selected at random subject to the condition that in each auxiliary pair x^j and y^j share a common subsequence at the same locations that primary x and y contain the *longest* common subsequence. However, we emphasize that the longest common subsequence for any of the auxiliary pairs, does not generally coincide with the longest common subsequence for the primary pair (x,y) . Therefore, we can not extract the solution to the LCS of (x,y) from a solution to one of the auxiliary pairs (x^j,y^j). Indeed, our solutions need $l=O(logn)$ auxiliary instances to be able to compute the solution to the primary instance and does not solve the LCS on auxiliary pairs (x^j,y^j). We also emphasize that the actual substring at the $\vec{a}$ locations of x^j does not agree with the substring at the $\vec{a}$ locations of the primary instance x, nor does the substring at the $\vec{b}$ locations of y^j match the substrings at the $\vec{b}$

locations of y, only the substring at locations $\vec{a}$ of x^j is required to match the substring at locations $\vec{b}$ of y^j.[1]

2.3 The New Algorithms in Exact and Relaxed Correlation Models

We start by describing our results for the LCS generating processeses.

Theorem 1 (LCS with exact correlation model) There exists an algorithm which on input $((x,y),(x^1,y^1),...,(x^t,y^t)$ generated according to the exact correlation model $GenLCS_{Pri}(m,t)$ defined above for Pri being a worst case distribution solves the LCS problem on string pair $(x,\ y\ )$ and runs in expected time $O(m\log m)$ for $t=O(\log m)$ where the expectation is taken over the choices of auxiliary inputs in $GenLCS_{Pri}(m,t)$.

The key technical insight for the aforementioned theorem is that it is now possible to construct a new instance of the LCS problem over an extended alphabet where each character corresponds to a vector in Σ^{l+1} and which now has enough structure (as opposed to the primary worst case instance) to be solved in sub-quadratic time. Essentially, each character in a position of the new LCS instance is determined by the characters in the original primary and auxiliary instances at the same position. When the number of auxiliary instances is large enough and the distribution is uniform, it will mean that the probability of any two non-matching positions having the same values for each of the auxiliary instances is small. This means that any positions that match on every auxiliary instance belong to the longest common subsequence with high probability. It is then possible to use hashing on the extended alphabet symbols to find the positions of the longest common subsequence.

The LCS results described are a special case of a general algorithm applied to any tuple-based similarity measure.

Main Theorem 2 (tuple-based similarity measures with exact correlation model) Let f be a tuple-based similarity measure on k strings of lengths $\vec{m}=m_1,...,m_k$ over alphabet Σ with no overlapping tuples. There exists an algorithm that finds $f(x_1,x_2,...,x_k)$ in time $\tilde{O}(kn\log n+|S|)$ where $|S|$ is the size of the largest allowed set of tuples on a primary instance $x_1,x_2,...,x_k$ and auxiliary instances $(x_1^{(1)},x_2^{(1)}...,x_k^{(1)}),...,(x_1^{(t)},...,x_k^{(t)})$ with $t=O(\log n)$ drawn from $Gen_{Pri,Aux}(f,k,\vec{m},t)$ with Pri worst-case and Aux uniform.
Note that in all of the similarity measures of interest, $|S|$ is of linear size in the input length.

Main Corollary 3

- On a primary instance $x_1,...,x_k$ of lengths $\vec{m}$ and $O(\log n)$, where $n=\max m_1,...,m_k$, auxiliary instances of k-LCS drawn from $Gen_{Pri,Aux}(LCS,k,\vec{m},O(\log n))$ with Pri worst-case and Aux uniform, we can find the longest common subsequence of the primary instance in time $\tilde{O}(nk)$.

[1] Interestingly, as we mentioned above, the induced distribution over the auxiliary instances produced by the exact correlation model, coincides with the semi-random model proposed by [3] in their work on the smooth complexity analysis of the EDIT problem. Recall, that they consider perturbing a worst case pair of strings (x,y) to (x',y') by randomly and independently choosing each of the characters of x' and y' in locations which were not part of the longest common subsequence of x and y; whereas for locations of x and y which are part of the longest common subsequence, x' and y' are forced to contain the same randomly chosen character. The existence of a linear time approximation algorithm for longest common subsequence for the (x',y') distribution, as shown by [3], does not seem to be useful directly for finding an approximate (nor exact) solution for LCS on (x,y) .

- On a primary instance x_1, x_2 of lengths m and n and $O(\log \max m, n)$ auxiliary instances of EDIT drawn from $Gen_{Pri,Aux}(EDIT, 2, \{m, n\}, O(\log \max m, n))$, we can find the minimum edit distance between the primary sequences in time $\tilde{O}(m + n)$.
- On a primary instance of x_1, x_2 of lengths m and n and $O(\log \max m, n)$ auxiliary instances of 0-1 DTWD drawn from $Gen_{Pri,Aux}(DTWD_{0,1}, 2, \{m, n\}, O(\log \max m, n))$ with Pri worst-case and Aux uniform, we can find the dynamic time warping distance of the primary instance in time $\tilde{O}(m + n)$.

Using considerably more complex algorithms and in particular techniques to find Hamming nearest neighbors, we extend the techniques to show the following for the relaxed correlation model.

Theorem 4 (LCS with relaxed correlation model) Let $\epsilon \in [0, \frac{1}{2}]$. There exists an algorithm which on a tuple of input instances $(x, y), (x^1, y^1), ..., (x^t, y^t)$ drawn from $GenLCS2_{Pri}(m, t, \epsilon)$ solves the LCS on string pair (x, y) with probability at least $2/3$ for $t = \Omega(\log m)$ and runs in expected time $O(tn^{1+d})$ for $d < 1$ (i.e sub quadratic time). (where the probability and runtime expectation is taken over the choices of auxiliary inputs in$GenLCS2_{Pri}(m, t, \epsilon)$.

Stating the general theorem for similarity measures.

Main Theorem 5 (tuple-based similarity measure with relaxed correlation model) Let f be a tuple-based similarity measure on k strings of lengths $\vec{m} = m_1, ..., m_k$ over alphabet Σ with no overlapping tuples. There exists an algorithm that finds $f(x_1, x_2, ..., x_k)$ in time $O(|S| + n^{2-\Omega(\epsilon^{1/3}/\log(1/\epsilon))})$ where $|S|$ is the size of the largest allowed set of tuples on a primary instance $(x_1, x_2, ..., x_k)$ and auxiliary instances $(x_1^{(1)}, x_2^{(1)} ..., x_k^{(1)}), ..., (x_1^{(t)}, ..., x_k^{(t)})$ drawn from $Gen_{Pri,Aux}(f, k, \vec{m}, t, \epsilon)$ with $t = O(\log n)$,Pri worst-case and Aux uniform.

Main Corollary 6

- On a primary instance $x_1, ..., x_k$ of lengths $\vec{m}$ and $O(\log n)$, where $n = \max m_1, ..., m_k$, auxiliary instances of k-LCS drawn from $Gen_{Pri,Aux}(LCS, k, \vec{m}, O(\log n), \epsilon)$ with Pri worst-case and Aux uniform, we can find the longest common subsequence of the primary instance in time $O(n^{2-\Omega(\epsilon^{1/3}/\log(1/\epsilon))})$.
- On a primary instance x_1, x_2 of lengths m and n and $O(\log \max m, n)$ auxiliary instances of EDIT drawn from
  $Gen_{Pri,Aux}(EDIT, 2, \{m, n\}, O(\log \max m, n), \epsilon)$, we can find the minimum edit distance between the primary sequences in time $O((m + n)^{2-\Omega(\epsilon^{1/3}/\log(1/\epsilon))})$.
- On a primary instance of x_1, x_2 of lengths m and n and $O(\log \max m, n)$ auxiliary instances of 0-1 DTWD drawn from $Gen_{Pri,Aux}(DTWD_{0,1}, 2, \{m, n\}, O(\log \max m, n), \epsilon)$ with Pri worst-case and Aux uniform, we can find the dynamic time warping distance of the primary instance in time $O((m + n)^{\Omega(\epsilon^{1/3}/\log(1/\epsilon))})$.

2.4 Another correlation model and results for EDIT

In addition to the correlation model implied for computing the min edit distance (EDIT), we consider another model which may be more natural to consider and show an efficient algoritme for EDIT in this model.

Consider the following generating process. Let $\pi_1, \pi_2, ..., \pi_k$ be the minimum sequence of edits needed to transform x to y. The auxiliary instances are random pairs of n-bit strings whose minimum edit sequence are the same as for the primary instance. Namely,

for auxiliary pair $x^{(1)}, y^{(1)}$, $y^{(1)}$ is obtained from $x^{(1)}$ using the same edit sequence which transformed x to y. Somewhat more generally, consider a worst case instance (x, y) and auxiliary instances $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ...$, with $x^{(i)} = x$ with each character changed with probability $\epsilon \in [0, 1]$ and $y^{(i)} = \pi_k(\pi_{k-1}(...(\pi_1(x^{(i)}))...))$. This is the distribution on which our algorithms perform.

Theorem 7(Solve EDIT exactly with $O(\log n)$ instances) There exists an algorithm A which on inputs generated as above solves the $EDIT(x, y)$ for a worst case primary instance and auxiliary instances as above such that on $O(\log n)$ auxiliary instances , the algorithm computes $EDIT(x, y)$ with high probability and runs in expected time $O(n \log n)$.

2.5 Overview of Techniques

The main idea of this paper is a new algorithm for solving similarity measure problems with correlated instances. This algorithm is significantly different than the dynamic programming algorithms that are used in the absence of correlated instances.

Given a tuple-based similarity measure, our goal is to find the set of tuples of locations which form the optimal (e.g. longest in LCS) solution to the primary instance $(x_1, ..., x_k)$. With correlated instances, consider a tuple of locations (e.g. $(i_1, ..., i_k)$ in $(x_1, ..., x_k)$), then in our exact correlation model, if these locations are not part of optimal solution these will be detected by comparing them across all instances, primary and auxiliary. This is the case as the correlation model makes it unlikely that even a pair of locations (i_u, i_v), which are not part of a matching tuple in the primary instance, would have the same value in the primary instances and all auxiliary instances.

Thus, algorithmically we can look at pairs of positions and determine whether they are in the same tuple by comparing them across all instances, primary and auxiliary. We do not need to do this for all pairs; we can use hashing to map pairs to each other, using the fact that similar pairs will be in the same tuple, and thereby find the tuples that determine the value of the similarity measure. The use of hashing proceeds as follows. First, we construct vectors out of the same position for the primary instance and each auxiliary instance for all of the strings. Then, we construct a hash table out of the vectors for the first string, and use the hash function to determine which vectors in the other strings are the same as vectors in the first string.

This idea works as is for the exact correlation model where pairs of locations in tuple of the optimal solution are guaranteed to contain the same value in all auxiliary instances, but fails when we relax this to the model where contents of such location pairs only have a higher probability of being the same.

In this case, if two locations were part of the same tuple, when we consider the vectors obtained from the auxiliary instances at those locations, they will be closer to each other than vectors obtained from positions which are not part of the same tuple. As a result, the nearest neighbors of the vector corresponding to a position are the vectors corresponding to the positions in the same tuple. Then, we can use algorithms that find nearest neighbors faster than brute-force search to find the tuples. Thus we can solve instances generated by the relaxed correlation model not in nearly linear time like the exact correlation model, but still significantly faster than the best known algorithms for the problem without correlated instances.

2.6 Related Works

We review the relevant results known in the literature for the EDIT,LCS, and DTWD problems.

Backurs and Indyk [4] have shown that if EDIT – the minimum number of insertions, deletions, and substitutions needs to transform one string into another – can be computed in time $O(n^{2-\delta})$ for some constant $\delta > 0$, then the satisfiability of CNF formulas with n variables and m clauses can be solved orm one string into anothertime $poly(m)2^{\epsilon n}$ for a constant $\epsilon > 0$. This would violate the Strong Exponential Time Hypothesis (SETH). A $O(n^{2-\delta})$ algorithm for EDIT would also imply a sub-quadratic algorithm for the orthogonal vector problem and a host of others.

Abboud, Backurs and Williams [1] addresses (among other problems) the LCS problem. They show that an $O(n^{2-\epsilon})$ algorithm for the LCS of two sequences of length n over a constant size alphabet would refute SETH and show a sub-quadratic algorithm for the orthogonal vectors problem and a host of others. For finding the longest subsequence common to k input sequences, it is similarly argued that achieving $O(n^{k-\epsilon})$ for any $\epsilon > 0$ is unlikely in [1].

Bringmann and Kunnermann [5] provide a lower bound for dynamic time warping distance, the minimum cost of a traversal of two strings where at each step, at least one of the strings moves forward, and the cost is the sum of the distances at each step. This problem is also solvable using dynamic programming, and [5] show that an $O(n^{2-\epsilon})$ that finds the dynamic time warping distance would refute SETH. Their proof uses the concept of an alignment gadget, which we will generalize with our notion of pair-based similarity measures.

We also mention work on the smoothed complexity of the EDIT distance problem. Recall, that the smoothed complexity of an algorithm by Spielmann and Teng [12], analyzes for an input x the expected behavior of the algorithm on correlated inputs which are the result of subjecting x to perturbations (e.g. flip its bits with a certain probability).

Andoni and Krauthgamer [3] study of the smoothed complexity of EDIT in the following perturbation model. Given two adversarially chosen binary input strings with common longest subsequence A , eacn character of the input strings is replaced independely (with some fixed probability p) with one restriction: the same perturbation is applied in both strings to the charcters in locations of the common longest subsequence A. They then show a constant factor approximation algorithms for EDIT which runs in nearly linear time for such perturbed instances.The existence of a linear time approximation algorithm for the LCS of (x', y') distribution, as shown by [3], does not seem to be useful directly for finding an approximate (nor exact) solution for the LCS of (x, y). Looking ahead, the stringest dostributions which we consider (which we call "exact correlation model") over auxiliary instances to a primary instance x essentially coincides with the [3] perturbation model applied to x.

We remark that the approach of smoothed analysis in general and the work of [3] in particuar, differs from our study and results in several aspects. First, in contrast to smoothed analysis our goal is not an analysis of when the problem becomes easier (or harder) but rather the development of algorithms which take advantage of (and when) extra correlated information is available in addition to an original input. Second, we solve the exact version of the min-EDIT distance problem on the primary worst case input pair itself with high probability (over the perturbed instances) whereas they approximate the min-EDIT distance on a perturbed instance.

Having said that, there is an interesting interplay between the explorations of smoothed analysis and improving algorithmics by access to correlated instances . That is, one may consider any distribution induced by semi-random models over problem instances both as

distributions for which smoothed complexity can be studied, or as distributions over the auxiliary correlated instances which are available for improving algorithms on a primary worst case instance.

Finally, we note that the idea of using larger alphabets to make the longest common subsequence problem easier is related to work of Hunt and Szymanski [9]. In this paper, they show that if r is the number of pairs of locations that are the same in the two sequences, there is an algorithm to compute the longest common subsequence of two strings of length n in time $O((r+n)\log n)$. Thus, if the number of matching pairs is small, the longest common subsequence can be computed easily. This idea is present in our work as well; when we consider correlated instances to make a new pair of sequences over a larger alphabet, the matching pairs will just be the pairs that are part of the longest common subsequence of the primary instance. Although the algorithm we use to find the longest common subsequence is different, the fundamental concept is the same.

2.7 Roadmap to The paper

The rest of the paper will proceed as follows. In Section 2, we will give the definition of a tuple-based similarity measure. We will also introduce the definitions of longest common subsequence, edit distance, and dynamic time warping distance and then show that LCS, EDIT, and a restricted version of DTWD are all tuple-based similarity measures. In Section 3, we define the correlation models used in the rest of the paper. We are then able to proceed to the results in Section 4. Section 4.1 covers the exact correlation model and Section 4.2 covers the relaxed correlation model. We leave the discussion of the other model for edit distance to the appendix.

3 A generalization of similarity measures

The longest common subsequence problem, edit distance problem, and dynamic time warping distance problem all have similar features which make them amenable to speedups using correlated instances. The feature that we will be looking at is the fact that the value of each of these similarity measures is the optimum over a set of values which only depend on the locations of the alignment. If this is the case, we will show that we can use a correlation model where the alignment stays the same to give a faster algorithm for finding this optimal alignment. The following definition provides a criterion which ensures that a similarity measure will have a good algorithm given correlated instances.

▶ **Definition 1.** *Let f be a function that takes k strings and outputs a number, which we think of as a measure of similarity between the strings. f is a tuple-based similarity measure on strings $x_1, ..., x_k$ if there exist $\mathcal{S} \subseteq \bigcup_{m_1,m_2,...,m_k} \mathcal{P}([m_1] \times [m_2] \times ... \times [m_k])$ of allowed sets of tuples, where $m_1, ..., m_k$ correspond to the length of each string, along with a function $c(S)$ computable in time $\tilde{O}(|S|)$ such that*
$f(x_1, x_2, ..., x_k) = \max_{S \in \mathcal{S}, \forall (i_1,i_2,...,i_k) \in S x_1[i_1]=x_2[i_2]=...=x_k[i_k]} c(S)$ *or*
$f(x_1, x_2, ..., x_k) = \min_{S \in \mathcal{S}, \forall (i_1,i_2,...,i_k) \in S x_1[i_1]=x_2[i_2]=...=x_k[i_k]} c(S)$.

▶ **Definition 2.** *A pair-based similarity measure is a tuple-based similarity measure for tuples of size 2.*

The motivation for this definition is to provide a generalization of the alignment gadget of [5] that captures the property we need to achieve faster algorithms given correlated instances. In [5], they introduce the notion of an alignment gadget, and show that any problem with an

alignment gadget cannot be solved in strongly subquadratic time. Our definition is more general than theirs in that we do not restrict ourselves to looking at alignments; multiple positions in one string can match to the same position in another string. However, we only look at measures where the function is determined by the positions of the matching pairs; this is for reasons related to the correlation model that will become apparent.

We denote the ith character of a string x over an alphabet Σ by $x[i]$.

▶ **Definition 3.** *A longest common subsequence of two strings $x, y \in \Sigma^*$ ($LCS(x, y)$) of length m is a largest set of pairs $(a_1, b_1), ..., (a_m, b_m)$ such that $1 \leq a_1 < a_2 < ... < a_m \leq n$, $1 \leq b_1 < b_2 < ... < b_m \leq n$, and $x[a_i] = y[b_i]$ for $i \in \{1, ..., m\}$.*

▶ **Definition 4.** *A longest common subsequence of k strings ($LCS(x[1], x[2], ..., x[k])$) of strings $x[1], x[2], x[3], ..., x[k]$ is a largest set of tuples $(a_1[1], a_2[1], ..., a_k[1]), ..., (a_1[m], a_2[m], ..., a_k[m])$ with $1 \leq a_j[1] < a_j[2] < ... < a_j[m] \leq n$ for all $j \in \{1, ..., k\}$ and $x_1[a_1[i]] = x_2[a_2[i]] = ... = x_k[a_k[i]]$ for all $i \in \{1, ..., m\}$*

It is easy to see that LCS is a tuple-based similarity measure when we take $c(S) = |S|$; the allowed sets of tuples are all possible alignments, and the maximum over all sets of alignments of $c(S)$ is precisely the length of the longest common subsequence.

▶ **Definition 5.** *The edit distance ($EDIT$) between two strings x, y is the minimum number of insertions, deletions, and character substitutions it takes to get from x to y. $EDIT(x, y) = LEV_{x,y}(|x|, |y|)$, where*

$$LEV_{x,y}(i,j) = \begin{cases} \max i, j & \text{if } \min i, j = 0 \\ \min \begin{cases} LEV_{x,y}(i, j-1) \\ LEV_{x,y}(i-1, j) \\ LEV_{x,y}(i-1, j-1) + \mathbb{1}_{a_i \neq b_j} \end{cases} \end{cases}$$

$$\text{and } \mathbb{1}_{a_i \neq b_j} = \begin{cases} 1 & \text{if } a_i \neq b_j \\ 0 & \text{if } a_i = b_j \end{cases}.$$

▶ **Proposition 1.** *$EDIT$ is a pair-based similarity measure.*

Proof. We observe that if every alignment has a unique minimum sequence edit associated with it, then the proposition follows as the set of pairs corresponding to a location unchanged by the minimum sequence of edits in the first string and the location where it moved to in the second string is an alignment, and thus taking the minimum over all alignments of the edit distance that preserves the alignment gives us the minimum edit distance. Thus it suffices to show that this is the case. Suppose that $(a_1, b_1), ..., (a_k, b_k)$ is the set of pairs that are unchanged when the edit operation is applied. We claim that the minimum number of edits leaving these pairs unchanged is equal to $\sum_{i=0}^{k} \max(a_{i+1} - a_i - 1, b_{i+1} - b_i - 1)$ if this is true. To see this, the edit takes $x[a_i]$ to $y[b_i]$, and thus everything between $x[a_i]$ and $x[a_{i+1}]$ must be matched to things between $y[b_i]$ and $y[b_{i+1}]$. This edit distance is equal to $\max(a_{i+1} - a_i - 1, b_{i+1} - b_i - 1)$. Thus, the minimum number of edits leaving this set of pairs unchanged is $\sum \max(a_{i+1} - a_i - 1, b_{i+1} - b_i - 1)$. Therefore the minimum edit distance is the minimum of this function over all sets of matching pairs that satisfy the constraint that for any i, j, either $a_i < a_j$ and $b_i < b_j$ or $a_i > a_j$ and $b_i > b_j$. ◀

▶ **Definition 6.** *A traversal of two sequences x, y of length m, n respectively is a list of pairs $(a_1, b_1), ..., (a_t, b_t)$ such that $(a_1, b_1) = (1, 1)$, $(a_t, b_t) = (m, n)$, and (a_{i+1}, b_{i+1}) is either $(a_i + 1, b_i), (a_i, b_i + 1)$, or $(a_i + 1, b_i + 1)$.*

▶ **Definition 7.** *The dynamic time warping distance (DTWD) between two strings x, y is the minimum cost of a traversal of the two strings. The cost of a traversal is the sum of the distances between each pair of the traversal.*

▶ **Proposition 2.** *DTWD when the distances are 0 and 1 ($DTWD_{0,1}$) is a pair-based similarity measure.*

Proof. Let $(a_1, b_1), ..., (a_k, b_k)$ be the set of pairs with $x[a_i] = y[b_i]$ in order in the traversal with the minimum cost. Similarly to the edit distance situation, the traversal must match everything between $x[a_i]$ and $x[a_{i+1}]$ and $y[b_i]$ and $y[b_{i+1}]$, and the dynamic time warping distance here is $\sum_{i=1}^{k-1} \max b_{i+1} - b_i - 1, a_{i+1} - a_i - 1$ because none of these were included in matching pairs because of the definition of a traversal. Then, determining the minimum dynamic time warping distance is equivalent to determining the minimum of this quantity over all subsets of traversals. This means that $DTWD_{0,1}$ is a pair-based similarity measure. ◀

4 Correlation models for tuple-based similarity measures

In this section we will define the way primary and auxiliary instances are generated by our generating processes. The first generating process draws primary instances from the primary distribution, and auxiliary instances from the auxiliary distribution conditioned on the optimal set of tuples for the primary instances matching for each auxiliary instance. This does not mean that the values at the locations specified by each tuple are the same between the primary and auxiliary instances; it indicates that the contents of the strings in an auxiliary instance at the locations indicated by the tuple will be the same. The second generating process will be similar to the first generating process, but instead of making the locations always match, there will be a higher probability of matching locations being the same.

▶ **Definition 8.** *Suppose that f is a tuple-based similarity measure with allowed sets of tuples $\mathcal{S}$ and cost function c. Let l be the number of correlated instances, k be the number of strings that f is a measure on, and let $m_1, ..., m_k$ be the length of each string.*

The generating process $Gen_{Pri,Aux}(f, k, \vec{m}, l)$ draws $x_1, x_2, ..., x_k$ of length $m_1, m_2, ..., m_k$ respectively from Pri with the allowed set of tuples $S = \{(a_{11}, a_{21}, ..., a_{k1}), ..., (a_{1m}, a_{2m}, ..., a_{km})\} = \arg\min_{S \in \mathcal{S} | \forall (i_1, i_2, ..., i_k) \in S, x_1[i_1] = x_2[i_2] = ... = x_k[i_k]} c(S)$. Then, it outputs l tuples $(x_1^{(i)}, x_2^{(i)}, ..., x_k^{(i)})$ drawn from $Aux^{\otimes n}$ conditioned on $\forall (j_1, j_2, ..., j_k) \in S, x_1^{(i)}[j_1] = x_2^{(i)}[j_2] = ... = x_k^{(i)}[j_k]$.

▶ **Definition 9.** *Suppose that f is a tuple-based similarity measure with allowed sets of tuples $\mathcal{S}$ and cost function c. Let l be the number of correlated instances, k be the number of strings that f is a measure on, ϵ be the probability that a given tuple is matching for an auxiliary instance, and let $m_1, ..., m_k$ be the length of each string.*

The generating process $Gen_{Pri,Aux}(f, k, \vec{m}, l, \epsilon)$ draws $x_1, x_2, ..., x_k$ of length $m_1, m_2, ..., m_k$ from Pri with the allowed set of tuples $S = \{(a_{11}, a_{21}, ..., a_{k1}), ..., (a_{1m}, a_{2m}, ..., a_{km})\} = \arg\min_{S \in \mathcal{S} | \forall (i_1, i_2, ..., i_k) \in S, x_1[i_1] = x_2[i_2] = ... = x_k[i_k]} c(S)$. Then, it outputs l tuples $(x_1^{(i)}, x_2^{(i)}, ..., x_k^{(i)})$ drawn from $Aux^{\otimes n}$ and then we make $x_1^{(i)}[j_1] = x_2^{(i)}[j_2] = ... = x_k^{(i)}[j_k]$ with probability ϵ for all tuples $(j_1, j_2, ..., j_k) \in S$.

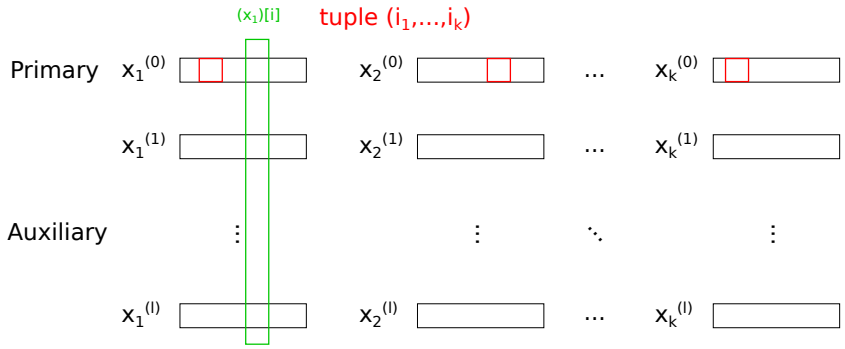


Figure 1 This shows the structure of the primary and auxiliary instances. The instance column is delineated in green, and a tuple in the primary instance is shown in red.

5 Results

5.1 Exact correlation model

What correlations should we expect to provide advantages? For tuple-based similarity measures, we should expect information about what the optimal set of tuples is to give us a speedup. In the model that we are considering, each location tuple in the optimal set of tuples of our primary instance will also be a set of locations with the same value in each auxiliary instance. For example, consider the two sequences 1010010 and 1101100. The longest common subsequence between the two sequences is 10110, so we can expect auxiliary instances to look like $abc^{**}de$ and $^{*}abcde^{*}$, where a, b, c, d, e have the same value and * is a wildcard.

This correlation model is not just solving a random instance of the problem. The goal is to find the optimal set of tuples for the primary instance. There are no guarantees on the distribution for the primary instance; the primary instance is worst-case. As a result, the problem is not equivalent to finding the optimal set of tuples for an auxiliary instance, and thus finding the correlations is not the same as solving a random instance of the original problem.

▶ **Definition 10.** *The instance column $(x)[i]$ is defined as follows. Suppose we have strings $(x_1, x_2, ..., x_k)$ which are primary instances, which we will refer to $(x_1^{(0)}, x_2^{(0)}, ..., x_k^{(0)})$, and auxiliary instances $(x_1^{(1)}, x_2^{(1)}, ..., x_k^{(1)})..., (x_1^{(l)}, x_2^{(l)}, ..., x_k^{(l)})$. Then, we denote $(x_j^{(0)}[i], x_j^{(1)}[i], ..., x_j^{(n)}[i]) = (x_j)[i]$ for any $j \in \{1, ..., k\}$.*

Given the correlation model, we know that two characters in each auxiliary instance will be equal with probability 1 if they belong to a tuple in the optimal set of tuples and with probability less than 1 otherwise. This suggests the following algorithm which runs in $O(kn \log n)$ time, or $O(n \log n)$ if k is constant. If we are given auxiliary instances from $Gen_{Pri,Aux}(f, k, \vec{m}, l)$ with Pri a worst-case distribution and Aux the uniform distribution, we can construct buckets indexed by elements of Σ_{l+1} and the entries of each bucket are the indices of the columns that have the same value as the bucket's index. Then, it looks up the columns of string 1 to see where the matches are. FIND proceeds in three steps; first, for the last $k-1$ strings, it puts each column of the string in the corresponding bucket in the set of buckets for that string. Next, it checks the columns of string 1 against the columns of string

2 to see which positions in string 1 are part of tuples. Then, we construct the set of tuples by looking up the positions in strings 2 through k that match to positions in string 1.

In FIND, the H_j are arrays of buckets where $((x_j)[i], i)$ is stored for $j = 2, ..., n$, and A is the array of k-tuples in the longest common subsequence. $H_j[s]$ denotes the contents of the bucket for the jth string, and is filled iff there exists i such that $(x_j)[i] = s$. $A_{i,j}$ is the index in the jth string of the ith entry of the longest common subsequence.

▶ **Theorem 1.** *Suppose we have a pair-based similarity measure or a k-tuple-based similarity measure with no overlapping tuples. If we are given a primary instance $x_1, x_2, ..., x_k$ and auxiliary instances $x_1^{(1)}, x_2^{(1)} ..., x_k^{(1)}, ..., x_1^{(l)}, ..., x_k^{(l)}$ with $l = O(\log n)$ drawn from $Gen_{Pri,Aux}(f, k, \vec{m}, l)$ with Pri worst-case and Aux uniform, FIND finds $f(x_1, x_2, ..., x_k)$ in time $\tilde{O}(kn \log n + |S|)$, where S is the largest set of tuples in $\mathcal{S}$ and $n = \max m_1, ..., m_k$.*

Proof. If A is the optimal set S^* from the primary instance, we can compute $f(x_1, x_2, ..., x_k) = c(A)$ in $\tilde{O}(|S^*|)$. Then, we will show that given the primary and auxiliary instances, the algorithm finds find the optimal set of tuples for the primary instance. Let $(i_1, ..., i_k) \in S^*$. By the definition of the correlation model, we will have that $(x_1)[i_1] = (x_2)[i_2] = ... = (x_k)[i_k]$ because the tuple is one of the optimal tuples in the primary instance which means that this is a matching tuple in both the primary and auxiliary instances. Suppose that we have a tuple $(i_1, i_2, ..., i_k)$ not in S^*. Then, the probability that $x_1^{(j)}[i_1] = x_2^{(j)}[i_2]$ for a given i is $1/2$, so then the probability of equality for all j is $\frac{1}{2^{O(\log n)}} < \frac{1}{3n^2}$. Taking a union bound over all pairs, we see that the probability of any tuple not in S^* having $(x_1)[i_1] = (x_2)[i_2]$ is at most $1/3$, so therefore with probability at least $2/3$, every tuple with $(x_1)[i_1] = (x_2)[i_2] = ... = (x_k)[i_k]$ will be be in S^*. Thus, with probability at least $2/3$, every hash collision corresponds to a tuple, and therefore A is the optimal set tuples for the primary instance. ◀

Algorithm 1: FIND$(f, x_1, x_2, ..., x_k, x_1^{(1)}, ..., x_k^{(1)}, ..., x_1^{(l)}, ..., x_k^{(l)})$

```
    Put columns into buckets
1   For j = 2, ..., k
2       Construct the array H_j with elements linked lists.
3       For i = 1, ...., n
4           Add ((x_j)[i], i) to H_j[(x_j)[i])].
5   Make the n × k array A.
6   m = 1
    Check which positions in string 1 are part of tuples in S*
7   For i = 1, ..., n
8       For (v, j) in H_2[(x_1)[i]]
9           If (x_1)[i] = v
10              A_{m,1} = i
11              A_{m,2} = j
12              m = m + 1
    Match positions in strings 2 through k to positions in string 1
13  For s = 3, ..., k
14      For l = 1, ..., m
15          i = A_{l,1}
16          If H_s[(x_1)[i]] is empty
17              Remove A_l and skip to the next l
18          For (v, j) in H_s[(x_1)[i]]
19              If (x_1)[i] = v
20                  A_{m,s} = j
21  Output c(A).
```

▶ **Corollary 1.** *If we are given a primary and $O(\log n)$, where $n = \max m_1, ..., m_k$, auxiliary instances of k-LCS drawn from $Gen_{Pri,Aux}(LCS, k, \vec{m}, O(k \log n))$ with Pri worst-case and Aux uniform, we can find the longest common subsequence of the primary instance in time $\tilde{O}(nk)$.*

▶ **Corollary 2.** *If we are given a primary and $O(\log \max m, n)$ auxiliary instance of EDIT drawn from $Gen_{Pri,Aux}(EDIT, 2, \{m, n\}, O(\log \max m, n))$, we can find the minimum edit distance between the primary sequences in time $\tilde{O}(m + n)$.*

▶ **Corollary 3.** *If we are given a primary and $O(\log \max m, n)$ auxiliary instances of $DTWD_{01}$ drawn from*
$Gen_{Pri,Aux}(DTWD_{0,1}, 2, \{m, n\}, O(\log \max m, n))$ with Pri worst-case and Aux uniform, we can find the dynamic time warping distance of the primary instance in time $\tilde{O}(m + n)$.

5.2 Relaxed correlation models

In the preceding section, we established that under a strict correlation model, we are able to find the solution to problems such as longest common subsequence and edit distance much faster than current lower bounds suggest if we did not have auxiliary instances. The main restriction is that the optimal set of tuples must also be a matching set of tuples for every auxiliary instance. When we talk about a matching tuple, we are referring to a tuple such that the characters at the locations specified by the tuple are the same. In this section, we will show how to relax this condition to give faster algorithms when a tuple in the optimal set of tuples only has a higher probability of being a matching tuple.

The main difficulty here is identifying the correct tuples. In the previous case, this was easy, because we could use the fact that every member of a correct tuple would be the same for each auxiliary instance. Then it sufficed to look at locations where the values matched up for each auxiliary instance, and we could use hashing to find these matching locations. In this case, we do not have this guarantee; we only have the guarantee that locations which are in the same tuple will have a higher probability of being the same in an auxiliary instance. This guarantee still allows us to find the elements of a tuple given its first element, as now the Hamming distance between two columns in the same tuple will be less than if they are not in the same tuple with high probability, and then we use faster algorithms for Hamming nearest neighbors such as the one in [2].

The following lemma will guarantee that the nearest neighbors of the column corresponding to the values of a position on the primary instance and auxiliary instances are the columns corresponding to the positions that are part of a tuple in the optimal set of tuples for the primary instance and has the first position in it also. As a result, this means that if we run a Hamming nearest neighbors algorithm on the columns, we will be able to find which tuples were the optimal tuples for the primary instance and thus compute the optimal value of the function for the primary instance.

▶ **Definition 11.** *$\mathbb{1}_{a=b}$ is 1 if $a = b$ and 0 otherwise.*

▶ **Lemma 1.** *Suppose we have x_1, x_2, ..., x_k and $x_1^{(1)}, x_2^{(1)}, ...x_k^{(1)}, ..., x_1^{(l)}, x_2^{(l)}, ..., x_k^{(l)}$ drawn from $Gen_{Pri,Aux}(f, k, \vec{m}, l\epsilon)$ with Pri worst-case and Aux uniform. Let $j \in \{2, ..., k\}, a \in \{1, ..., m_1\}$ and $b \in \{1, ..., m_j\}$. Then, for any δ such that $0 < \delta < 1/2 + \varepsilon$, if $a = i_1$ and $b = i_j$ for some $(i_1, ..., i_k) \in S^*$, the optimal set of tuples for the primary instance,*

$$\Pr[\mathbb{E}_m[\mathbb{1}_{x_1^{(m)}[a]=x_j^{(m)}[b]}] \leq 1/2 + \varepsilon - \delta] \leq e^{-\frac{(\delta/(1/2+\varepsilon))^2 l}{2}}$$

and otherwise

$$\Pr[\mathbb{E}_m[\mathbb{1}_{x_1^{(m)}[a]=x_j^{(m)}[b]}] \geq 1/2+\delta] \leq e^{-4\delta^2 l/3}$$

Proof. We use the form of the Chernoff bound given in [11]: for X a sum of independent random variables $X_1, ..., X_n$ taking values $\{0,1\}$ with expectation μ, we have that $\Pr[X \leq (1-\delta)\mu] \leq e^{-\delta^2\mu/2}$ and $\Pr[X \geq (1+\delta)\mu] \leq e^{-\delta^2\mu/3}$ for $0 < \delta < 1$. The first inequality is proven by plugging in $\mu = (1/2+\varepsilon) * l$ into the first Chernoff bound and noting that the expectation is the sum of the indicator random variables divided by l, and the second inequality follows similarly. ◀

Given this result, we can now use an algorithm to find Hamming nearest neighbors to find the optimal set of tuples. The best algorithm is the one of [2], which runs in time $n^{2-\Omega(\epsilon^{1/3}/\log(1/\epsilon))}$ to find the approximate nearest neighbors. What follows is the algorithm, treating finding the Hamming nearest neighbors as a black box.

FIND-RELAXED$(f, x_1, x_2, ..., x_k, x_1^{(1)}, ..., x_k^{(1)}, ..., x_1^{(l)}, ..., x_k^{(l)}, \epsilon)$

1 Make an $n \times k$ array A.
2 For $i = 1, ..., m_1$
3 For $j = 2, ..., k$ and $t = 1, ..., m_k$, find all vectors $(x_j)[t]$ that are less than $(1/2 - \epsilon/2) * l$ away from $(x_1)[i]$ in Hamming distance using the algorithm for approximate nearest neighbors
4 Add the tuple of all these vectors to A
5 Compute $c(A)$.

▶ **Theorem 2.** *Suppose we have a pair-based similarity measure or a k-tuple-based similarity measure with no overlapping tuples f. Let $l = 1000 \log n$. With probability at least $2/3 - o(1)$, the preceding algorithm computes $f(x_1, ..., x_k)$ given correlated instances from $Gen_{Pri,Aux}(f, k, \vec{m}, l, \epsilon)$ with Pri worst-case and Aux uniform in time $O(|S| + n^{2-\Omega(\epsilon^{1/3}/\log(1/\epsilon))})$, where S is the largest set of tuples in $\mathcal{S}$ and $n = \max m_1, ..., m_k$.*

Proof. Given $1000 \log n$ instances, we can use the lemma to say that for any two columns, with probability $1/n^3$ they do not correspond to a tuple and have Hamming distance less than $1/2 - \epsilon/4$ or they do correspond to a tuple and have Hamming distance more than $1/2 - 3\epsilon/4$. Taking a union bound gets us that the probability that this holds for any pair of columns is less than k/n. This means that the nearest neighbors of every column are the columns that it shares a tuple with, with probability $1 - o(1)$. Then in time $O(n^{2-\Omega(\epsilon^{1/3}/\log(1/\epsilon))})$ using the algorithm of [2] we can find the optimal set of tuples S^* with probability $2/3$. Then in time $|S|$ we can compute $c(S^*) = f(x_1, ..., x_k)$. ◀

▶ **Corollary 4.** *If we are given a primary and $O(\log n)$, where $n = \max m_1, ..., m_k$, auxiliary instances of k-LCS drawn from $Gen_{Pri,Aux}(LCS, k, \vec{m}, O(k \log n), \epsilon)$ with Pri worst-case and Aux uniform, we can find the longest common subsequence of the primary instance in time $O(n^{2-\Omega(\epsilon^{1/3}/\log(1/\epsilon))})$.*

▶ **Corollary 5.** *If we are given a primary and $O(\log \max m, n)$ auxiliary instance of EDIT drawn from $Gen_{Pri,Aux}(EDIT, \{m, n\}, 2, O(\log \max m, n), \epsilon)$, we can find the minimum edit distance between the primary sequences in time $O(n^{2-\Omega(\epsilon^{1/3}/\log(1/\epsilon))})$.*

▶ **Corollary 6.** *If we are given a primary and $O(\log \max m, n)$ auxiliary instances of $DTWD_{01}$ drawn from $Gen_{Pri,Aux}(DTWD_{01}, \{m, n\}, 2, O(\log \max m, n), \epsilon)$ with Pri worst-case and Aux uniform, we can find the dynamic time warping distance of the primary instance in time $O(n^{2-\Omega(\epsilon^{1/3}/\log(1/\epsilon))})$.*

6 Another correlation model for edit distance

▶ **Definition 12.** *The generating process* $Genedit_{Pri,Aux,\epsilon}(n,l)$ *draws* x, y *of length* n *and a sequence of character insertions, deletions, and substitutions* $\pi_1, \pi_2, ..., \pi_k$ *that is the minimum sequence of edits needed to transform* x *to* y *from* Pri*. The output is* $x, y, x^{(1)}, y^{(1)}, x^{(2)}, y^{(2)}, ..., x^{(l)}, y^{(l)}$*, with* $x^{(i)}$ *being* x *with each character changed with probability* ϵ *and* $y^{(i)} = \pi_k(\pi_{k-1}(...(\pi_1(x^{(i)}))...))$.

Suppose that we have two strings x, y for which we want to compute the minimum edit distance. If our auxiliary instances have the same sequence of edit operations, then with $3 \log n$ auxiliary instances the probability that $(x)_i = (y)_j$ and x_i did not to y_j during the sequence of inserts, deletes, and changes is less than $1/n^3$. This means that we can find the parts of the original string that are preserved under the edit and their new positions. Our algorithm works as follows. First, we get $O(\log n)$ auxiliary instances from $Genedit_{Pri,Aux,\epsilon}$, where Pri is worst-case, and then put the $(y)_j$ in buckets and check which $(x)_i$ have $(x)_i = (y)_j$. Then, with the pairs $(a_1, b_1), ..., (a_k, b_k)$, the edit distance is $\sum_{i=0}^{k} \max a_{i+1} - a_i - 1, b_{i+1} - b_i - 1$ with high probability. This algorithm obviously runs in $O(n \log n)$ time.

In FIND-EDIT, the array H of buckets is used to store $(y)_i$, and $COMMONPAIRS$ holds the i, j such that $(x)_i = (y)_j$.

Algorithm 2: FIND-EDIT$(x, y, x^{(1)}, y^{(1)}, ..., x^{(l)}, y^{(l)})$

```
1  Initialize an n × 2 array COMMONPAIRS.
2  Construct an array H of size n of linked lists.
3  For j = 1, ..., n
4        Add ((y)_j, j) to H[(y)_j].
5  For i = 1, ..., n
6        For (v, j) in H[(x)_i]
7              If (x)_i = v add (i, j) to COMMONPAIRS.
8  Output Σ_{i=0}^{k} max (a_{i+1} − a_i − 1, b_{i+1} − b_i − 1),
   where ((a_1, b_1), (a_2, b_2), ..., (a_k, b_k)) = COMMONPAIRS.
```

▶ **Claim 1.** *The algorithm* FIND-EDIT *on* $O(\log n)$ *instances generated by* $Genedit_{Pri,Aux,\epsilon}(n,l)$ *with* Pri *worst-case and* Aux *uniform works and runs in time* $O(n \log n)$ *with probability at least 7/8.* [2]

Proof. Suppose that x_i did not move to y_j by the original sequence of edits. Then, the probability that $x_i^{(k)} = y_j^{(k)}$ is $1/2$ because they were drawn independently uniformly at random. Suppose we have $2 \log n + 3$ auxiliary instances. This means that the probability that $(x)_i = (y)_j$ is at most $1/8n^2$ because each auxiliary instance is independent. Taking a union bound means that the probability that there is a pair x_i, y_j where x_i does not correspond to y_j and $(x)_i = (y)_j$ is at most '/8. Thus, the hashing finds the pairs $(a_1, b_1), ..., (a_k, b_k)$ with probability at least 7/8 in expected time $O(n \log n)$. We claim that the edit distance is equal to $\sum_{i=0}^{k} \max (a_{i+1} - a_i - 1, b_{i+1} - b_i - 1)$ if this is true. To see this, the original edit takes x_{a_i} to y_{b_i}, and thus everything between x_{a_i} and $x_{a_{i+1}}$ must be matched to things between y_{b_i} and $y_{b_{i+1}}$. This edit distance is equal to $\max (a_{i+1} - a_i - 1, b_{i+1} - b_i - 1)$. Suppose that the edit distance was less. Then there must be a coordinate x_c or y_d that was in the original string.

[2] Recall that $Genedit_{Pri,Aux,\epsilon}$ outputs auxiliary instances where the first string is a perturbation of the original string and the second string is obtained by applying the same edits to the first string.

But then, we could extend this to a smaller edit for the entire string than the original edit, which is a contradiction. Thus, the edit distance is at least $\max(a_{i+1} - a_i - 1, b_{i+1} - b_i - 1)$, and this is achieved by the original edit, because none of them are preserved. The running time is $O(n \log n)$ because each loop takes time $O(n \log n)$. ◀

Acknowledgements. We are grateful to Guy Rothblum for early important discussions on this work and the choices of correlation models. Thank you Guy! We also thank Aviv Regev for helpful discussion on correlations in sequence alignment in bioinformatics.

References

1 Amir Abboud, Arturs Backurs, and V. Vassilevska Williams. Tight hardness results for lcs and other sequence similarity measures. In *FOCS 2015*, 2015.

2 Josh Alman, Timothy M Chan, and Ryan Williams. Polynomial representations of threshold functions and algorithmic applications. *arXiv preprint arXiv:1608.04355*, 2016.

3 Alexandr Andoni and Robert Krauthgamer. The smoothed complexity of edit distance. *ACM Transactions on Algorithms (TALG)*, 8(4):44, 2012.

4 Piotr Indyk Arturs Backurs. Edit distance cannot be computed in strongly subquadratic time (unless seth is false). In *STOC15*, 2015.

5 Karl Bringmann and Marvin Künnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 79–97. IEEE, 2015.

6 Irit Dinur, Shafi Goldwasser, and Huijia Lin. The computational benefit of correlated instances. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 219–228, 2015. doi:10.1145/2688073.2688082.

7 Anka Gajentaan and Mark H Overmars. On a class of o(n^2) problems in computational geometry. *Computational geometry*, 5(3):165–185, 1995.

8 Nadia Heninger, Zakir Durumeric, Eric Wustrow, and J Alex Halderman. Mining your ps and qs: Detection of widespread weak keys in network devices. In *Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)*, pages 205–220, 2012.

9 James W Hunt and Thomas G Szymanski. A fast algorithm for computing longest common subsequences. *Communications of the ACM*, 20(5):350–353, 1977.

10 Russell Impagliazzo and Ramamohan Paturi. Complexity of k-sat. In *Computational Complexity, 1999. Proceedings. Fourteenth Annual IEEE Conference on*, pages 237–240. IEEE, 1999.

11 Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

12 Daniel A Spielman and Shang-Hua Teng. Smoothed analysis. In *Algorithms and data structures*, pages 256–270. Springer, 2003.

Multi-Clique-Width*

Martin Fürer†

The Pennsylvania State University, University Park, USA
furer@cse.psu.edu

Abstract

Multi-clique-width is obtained by a simple modification in the definition of clique-width. It has the advantage of providing a natural extension of tree-width. Unlike clique-width, it does not explode exponentially compared to tree-width. Efficient algorithms based on multi-clique-width are still possible for interesting tasks like computing the independent set polynomial or testing c-colorability. In particular, c-colorability can be tested in time linear in n and singly exponential in c and the width k of a given multi-k-expression. For these tasks, the running time as a function of the multi-clique-width is the same as the running time of the fastest known algorithm as a function of the clique-width. This results in an exponential speed-up for some graphs, if the corresponding graph generating expressions are given. The reason is that the multi-clique-width is never bigger, but is exponentially smaller than the clique-width for many graphs. This gap shows up when the tree-width is basically equal to the multi-clique width as well as when the tree-width is not bounded by any function of the clique-width.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.2 Graph Theory, D.2.8 Metrics

Keywords and phrases clique-width, parameterized complexity, tree-width, independent set polynomial, graph coloring

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.14

1 Introduction

Tree-width is the first and by far the most important width parameter. It is motivated by the fact that almost all interesting problems that are hard for general graphs allow efficient algorithms when restricted to trees. Furthermore such algorithms are often quite trivial. The promise of the notion of tree-width is to extend such efficient algorithms to much larger classes of tree-like graphs. Graphs of bounded tree-width have one shortcoming though. They are all sparse.

Clique-width [8] is the second most important width parameter. It has been defined by Courcelle and Olariu [12] based on previously used operations [8]. It is intended to compensate for the main shortcoming of the class of graphs of bounded tree-width. The idea is that many graphs are not sparse, but are still constructed in a somewhat simple and uniform way. One would expect to find efficient algorithms for such graphs too. The most extreme example is the clique. It's hard to find a natural problem that is difficult for a clique.

It turns out that graphs of bounded tree-width actually also have bounded clique-width [12, 6], and many efficient algorithms extend to the larger class. Indeed, every graph property expressible in $\mathcal{MS}_1$, the monadic second order logic with set quantifiers for vertices only,

* This work was partially supported by NSF Grant CCF-1320814.
† Part of this work has been done while visiting Theoretical Computer Science, ETH Zürich, Switzerland


© Martin Fürer;
licensed under Creative Commons License CC-BY

is decidable in linear time (linear in $|V| + |E|$) for graphs of bounded clique-width [10]. On the other hand, graph properties expressible in $\mathcal{MS}_2$, the monadic second order logic with set quantifiers for vertices and edges, is decidable in linear time for graphs of bounded tree-width [7], but not for graphs of bounded clique-width [10] under suitable complexity assumptions.

Recently the split-matching-width (sm-width), a new width parameter has been defined [25] to handle some of the $\mathcal{MS}_2$ expressible problems for unbounded tree-width. The strength of sm-width is strictly between tree-width and clique-width. We will not discuss it, because we focus on parameters that are at least as strong as clique-width.

We are concerned with the fact that the containment of the bounded tree-width graphs in bounded clique-width graphs is not obvious. Furthermore, the generalization from bounded tree-width to bounded clique-width does not come cheap. The width can blow up exponentially, with a potential for a significant loss of efficiency for many algorithms.

This creates a cumbersome situation for the many problems that have efficient solutions in terms of tree-width as well as in terms of clique-width, assuming the corresponding decompositions are known. One would like to run the algorithm based on clique-width to cover a much larger class of graphs, but that would mean an exponential sacrifice in running time for some graphs with small tree-width. Arguably, there should be a notion of a width parameter that bridges this gap more graciously.

Naturally, this aesthetic goal is not the whole story. One could run a tree-width and a clique-width based algorithm and take the result from the one that finishes first. Maybe the following argument is more important. If every clique-width based algorithm is exponentially slower than a tree-width based algorithm for a typical graph of small tree-width, then this seems to indicate some deficiency of the notion of clique-width. Therefore, what we really want, and what we will achieve, is finding a natural notion of width, such that for many important computational problems, algorithms based on this width have all of the following desirable properties.

- The algorithms based on this width are at least as fast as algorithms based on tree-width.
- The algorithms based on this width are at least as fast as algorithms based on clique-width.
- The algorithms based on this width are exponentially faster than algorithms based on clique-width, not only for graphs of small tree-width, but for many graphs of arbitrary large tree-width.

The current author has searched for some time for a parameter naturally generalizing tree-width and clique-width. Ideally, there should be no exponential blow-up in the parameter value. The second objective has been obtained with the notion of fusion-width [18]. It has been shown before that the fusion operation does not produce unbounded clique-width graphs from bounded clique-width graphs [9]. Indeed, there is a much tighter relationship. Graphs of tree-width k have fusion-width at most $k + 2$ [18], while in the worst case, they have clique-width exponential in k [6].

This is a very desirable property of fusion width. The drawback is that attaching a fusion operation is somewhat unnatural. It is an artificial push of the tree-width concept into a clique-width-like environment. Here, we investigate a far more natural width parameter that achieves this goal in a more direct way. We call it *multi-clique-width*, it is obtained by a simple modification in the notion of clique-width, namely by allowing every vertex to have multiple labels. Furthermore, it seems that the multi-clique-width might often be exponentially smaller than the fusion-width. On the other hand, it can be shown that multi-clique-width is never more than twice the fusion-width.

In this paper, we propose multi-clique-width as a serious contender of clique-width. This powerful parameter has some very desirable properties. Its definition is equally simple and natural as that of clique-width. The multi-clique-width is never bigger than the clique-width, but often exponentially smaller [13]. And most importantly, there is no explosion of the width when moving from tree-width to multi-clique-width. Furthermore, there are interesting algorithms where the dependence of the running time on the (potentially much smaller) multi-clique-width is about the same as the dependence on the clique width for a similar algorithm working with clique-width. Thus, multi-clique-width allows some tasks to be solved much more efficiently than previously known.

There are other important width parameter, the rank-width [22] and the boolean-width [5], that share some significant properties with the multi-clique-width. They too are never bigger than the clique-width and can be exponentially smaller. So why do we want to investigate yet another similar parameter?

Rank-width, boolean-width, clique-width, and multi-clique-width are all equivalent in the sense that the exact same problems are solvable in polynomial time for bounded width. If one of these parameters is bounded (by a constant), then so are the others. Rank-width has been introduced with this equivalence in mind [22]. Before, the graphs of bounded clique-width could not be identified computationally. Therefore, graphs of bounded clique-width could only be handled efficiently, when a corresponding k-expression had been given. Now the rank-width $\mathrm{rw}(G)$ can be approximated by a constant factor, and a $2^{3\mathrm{rw}(G)+2}-1$-expression can be computed in time $O(f(k)\,n^9 \log n)$ for some function f [22]. The later algorithm [20] to compute the rank-width exactly with a similar running time has not much effect on the approximability of clique-width. In theoretical investigations, the exponential bound on the clique-width has not often been viewed as a major concern, because the main goal has been to handle bounded clique-width graphs in polynomial time, not to speed them up, even when the rank-width is much smaller than the clique-width.

There are algorithms based directly on rank-width [19]. Still, for many application algorithms, clique-width based algorithms are more natural and easier to design.

We will show that for some computational tasks multi-clique-width can be used with the same ease as clique-width, but with an exponential speed-up for many graphs. These are exponential speed-ups in the parameter, meaning that the class of graphs with bounded parameter value would not change, just the computations get much faster.

Many questions are not yet answered for multi-clique-width. We conjecture multi-clique-width to be NP-hard. Nevertheless it might be that multi-clique-width could be approximated by a constant factor in polynomial time. We don't yet know. These are open problems for clique-width and boolean-width too.

In comparison with boolean-width or rank-width, when the corresponding decompositions are given, we notice that the known efficient algorithms for NP-complete problems are quadratic in n for bounded width, while for $\mathcal{MS}_1$ expressible graph properties, we have linear time algorithms [10] for bounded clique-width or multi-clique-width. Furthermore, we will illustrate that such algorithms can be quite simple and efficient as a function of the multi-clique-width.

Naturally, one could argue that this is an unfair comparison. The linear time is a result of the structural information that is given by the expression defining a graph with a bound on the clique-width or multi-clique-width. Boolean-width or rank-width on the other hand are defined with existential properties that constantly require some search in the graph. There are no concise boolean expressions or rank expressions defining these graphs. There is truth in this argument, but at the same time, one could claim that this is still an advantage of clique-width and multi-clique width. Maybe, some nice classes of graphs have

nice expressions describing them. If such expressions are known or can easily be found, they certainly should be used for efficient algorithms.

2 Definitions and Preliminaries

We use the standard notions of tree decomposition, tree-width, k-expression, and clique-width.

▶ **Definition 1.** A *tree decomposition* of a graph $G = (V, E)$ is a pair $(\{B_i : i \in I\}, T)$, where $T = (I, F)$ is a tree and each node $i \in I$ has a subset $B_i \subseteq V$ of vertices (called the bag of i) associated to it with the following properties.

1. $\bigcup_{i \in I} B_i = V$, i.e., each vertex belongs to at least one bag.
2. For all edges $e = \{p, q\} \in E$, there is at least one $i \in I$ with $\{p, q\} \subseteq B_i$, i.e., each edge is represented by at least one bag.
3. For every vertex $v \in V$, the set of indices i of bags containing v induces a subtree of T (i.e., a connected subgraph of T).

▶ **Definition 2.** The *width of a tree decomposition* is 1 less than its largest bag size. The *tree-width* $\mathrm{tw}(G)$ [24] of a graph G is the width of a minimal width tree decomposition of G.

It is NP-complete to decide whether the tree-width of a graph is at most k (if k is part of the input) [1]. For every fixed k, there is a linear time algorithm deciding whether the tree-width is at most k, and if that is the case, producing a corresponding tree decomposition [2]. For arbitrary k, this task can still be approximated. A tree decomposition of width $O(k \log n)$ can be found in polynomial time [4], and in time $O(c^k n)$ almost a 5-approximation [3] can be found (a tree of width at most $5k + 4$ to be precise). Furthermore, a tree decomposition of width $O(k^2)$ can be found in time $O(k^7 n \log n)$ [17].

It is often convenient to view T as a rooted tree, where an arbitrary fixed node has been chosen as the root.

▶ **Definition 3.** A *semi-smooth tree decomposition* of width k is a rooted tree decomposition where the bag B_i of every node i contains exactly 1 vertex that is not in the bag of the parent node $p(i)$. For rooted trees T with $v \in B_i \setminus B_{p(i)}$, we say that *node i is the* home *of vertex v*.

In other words, the home of a vertex v is the highest node whose bag contains v. The bag B_r of the root r of a semi-smooth tree decomposition contains just one vertex.

▶ **Proposition 4.** *Every graph $G = (V, E)$ has a semi-smooth tree decomposition of width $k = \mathrm{tw}(G)$ with $|I| = |V|$. Any tree decomposition of bounded width can be transformed into a semi-smooth tree decomposition in linear time.*

Proof. Do a depth-first search of the tree and omit nodes whose bag is contained in the bag of the parent. Insert intermediate nodes if more than one vertex has the same home. ◀

We use the standard notation of k-expression to define clique-width.

▶ **Definition 5.** A *k-expression* is an expression formed from the atoms $i(v)$, the two unary operations $\eta_{i,j}$ and $\rho_{i \to j}$, and one binary operation $\oplus$ as follows.

- $i(v)$ creates a vertex v with label i, where i is from the set $\{1, \ldots, k\}$.
- $\eta_{i,j}$ creates an edge between every vertex with label i and every vertex with label j for $i \neq j$ (with $i, j \in \{1, \ldots, k\}$).
- $\rho_{i \to j}$ changes all labels i to j (with $i, j \in \{1, \ldots, k\}$).
- $\oplus$ (join-operation) does a disjoint union of the generated labeled graphs.

Finally, the *generated graph* is obtained by deleting the labels.

We also allow multi-way join-operations, as $\oplus$ is associative.

▶ **Definition 6.** The *clique-width* $\mathrm{cw}(G)$ of a graph is the smallest k such that the graph can be defined by a k-expression [12].

Computing the clique-width is NP-hard [15]. Thus, one usually assumes that a graph is given together with a k-expression.

Theoretically, this is not necessary, because for constant k, the clique-width can be approximated in polynomial time [22, 21]. But the approximation ratio is exponential in k.

Now we define multi-clique-width in a similar way as clique-width. The essential difference is that every vertex can have any set of labels (including singleton sets and the empty set). There is a new operation ϵ_i to delete a label. The creation of multiple vertices with the same set of labels by one command is an unessential convenience.

▶ **Definition 7.** A *multi-k-expression* is an expression formed from the atoms $m\langle i_1, \ldots, i_\ell\rangle$, the three unary operations $\eta_{i,j}$, $\rho_{i\to S}$, and ϵ_i, as well as the binary operation $\oplus$ as follows. Assume $i, j \in \{1, \ldots, k\}$, $\ell \in \{0, \ldots, k\}$ and $\emptyset \subseteq S, \{i_1, \ldots, i_\ell\} \subseteq \{1, \ldots, k\}$).

- $m\langle i_1, \ldots, i_\ell\rangle$ with m a positive integer and $i_1 < \cdots < i_\ell \leq k$, creates m vertices, each with label set $\{i_1, \ldots, i_\ell\}$.
- $\eta_{i,j}$ creates an edge between every vertex u with label i and every vertex v with label j. This operation is only allowed when there are no vertices with label i and j simultaneously, in particular $i \neq j$.
- $\rho_{i\to S}$ replaces label i by the set of labels S, i.e., if a vertex v had label set S' with $i \in S'$ before this operation, then v has label set $(S' \setminus i) \cup S$ after the operation.
- ϵ_i deletes the label i from all vertices.
- $\oplus$ (join-operation) does a disjoint union of the generated labeled graphs.

Finally, the *generated graph* is obtained by deleting the labels.

S and $\{i_1, \ldots, i_\ell\}$ are allowed to be empty, even though the latter is not very interesting, as it only creates isolated vertices. Note that ϵ_i is just the special case of $\rho_{i\to S}$ with $S = \emptyset$. We list it separately, because one might want to consider the *strict multi-k-expressions* without $\rho_{i\to S}$. In Theorem 13 below, $\rho_{i\to S}$ is not used. Alternatively, one might restrict $\rho_{i\to S}$ to the classical case with S being a singleton. The relative power of these 3 versions might be worth studying.

▶ **Definition 8.** The *multi-clique-width* $\mathrm{mcw}(G)$ of a graph is the smallest k such that the graph can be defined by a multi-k-expression.

It turns out that multiply labeled graphs have been used before [12] in a more auxiliary role, and a variant of the multi-clique-width has actually appeared in the literature under the name m-clique-width [13] in the context of preprocessing for shortest path routing computations.

Here we list the standard definition of boolean-width in order to compare it with multi-clique-width.

▶ **Definition 9.** A *decomposition tree* of a graph $G = (V, E)$ is a tree T where V is the set of leaves and where all internal nodes have degree 3.
Every edge e of T defines a partition of V into X and $\overline{X}$ consisting of the leaves of the two trees obtained from T by removing e.
The *set of unions of neighborhoods* of X across the cut $\{X, \overline{X}\}$ is the set

$$U(X) = \{S' \subseteq \overline{X} \mid \exists S \subseteq X \;\; S' = N(S) \cap \overline{X}\}.$$

X defines bool-dim$(X) = \log_2 |U(X)|$.
The *boolean-width* of G is the minimum over all trees T of the maximum over all cuts $\{X, \overline{X}\}$ defined by an edge e of T of bool-dim$(X) = \log_2 |U(X)|$.

3 Relationship between Different Width Parameters

Multi-clique-width extends the notions of tree-width and of clique-width in a natural way.

▶ **Proposition 10.** *For every graph G,* $\mathrm{mcw}(G) \leq \mathrm{cw}(G) \leq 2^{\mathrm{mcw}(G)}$.

Proof. The first inequality directly follows from the definitions. For the second inequality, just use a label for every set of labels. ◀

▶ **Corollary 11.** *A class of graphs has bounded clique-width if and only if it has bounded multi-clique-width.*

▶ **Corollary 12.** *Properties of graphs expressible in $\mathcal{MS}_1$ (monadic second order logic without quantifiers over sets of edges) are linear time decidable for graphs of bounded multi-clique-width.*

Proof. This follows from Corollary 11 and the corresponding meta-theorem for clique-width [10]. ◀

▶ **Theorem 13.** *If a tree-decomposition of width k of a graph $G = (V, E)$ is given, then a multi-$(k+2)$-expression for G can be found in polynomial time.*

Proof. Assume, G is given with a tree decomposition of width $k = \mathrm{tw}(G)$. In linear time, the tree decomposition is transformed into a semi-smooth tree decomposition $(\{B_i : i \in I\}, T)$. Now we assign an identifier $\iota(v)$ from $\{1, 2, \ldots, k+1\}$ to each vertex v top-down, i.e., starting at the root of T. When identifiers have been assigned to the vertices whose homes are above the home of vertex v, we assign to vertex v the smallest identifier not assigned to the other vertices in the bag of the home of v.

Next, we define a multi-$(k+2)$-expression whose parse tree T' is basically isomorphic to the tree T of the tree decomposition. The difference it that in T' every internal node has an additional child that is a leaf. We call it an auxiliary leaf. Furthermore, above each internal node i, we introduce three auxiliary nodes obtained by subdividing the edge to the parent of i.

The main idea is that every vertex v is created at its home, or more precisely, in the auxiliary node below its home. Then the edges from v to neighbors of v with a home further down the tree are added. The upper neighbors of v, i.e., those that have their home higher up the tree, are not yet created. Vertex v remembers to attach to these neighbors later by taking the set of identifiers of these neighbors as its labels. All upper neighbors of v are together with v in the bag B_i of the home i of v in T. The vertex v needs at most k labels for this purpose. We give v an additional label, $k+2$, to allow the lower neighbors of v to connect to v. Node i of T' is a multi-way join operation of all its children, including the new auxiliary child. The purpose of the three nodes inserted above node i is to add the edges between v and its neighbors in the subtree of i, and to delete the two labels ($k+2$, and $\iota(v)$, the identifier of v) that have been used to create these new edges. The multi-$(k+2)$-expression is built bottom-up.

Now we define the multi-$(k+2)$-expression exactly by assigning atoms to the leaves and operations to the internal nodes as follows.

Regular leaf: Let the leaf i be the home of some vertex v. Let $v_1, \ldots, v_\ell$ be the neighbors of v with identifiers $i_1, \ldots, i_\ell$. Clearly, $\{v, v_1, \ldots, v_\ell\} \subseteq B_i$. Then the expression $1\langle i_1, \ldots, i_\ell\rangle$ creates v in leaf i.

Auxiliary leaf: Let the internal node i be the home of some vertex v. Let $v_1, \ldots, v_\ell$ be the upper neighbors of v with identifiers $i_1, \ldots, i_\ell$. Let $c_0(i)$ be the child of i which is an auxiliary leaf. Then the expression for $c_0(i)$ is $1\langle k+2, i_1, \ldots, i_\ell\rangle$.

Internal node: Let i be the home of some vertex v, and let $c_1(i), \ldots, c_q(i)$ be the children of i in T. Let $c_0(i)$ be the auxiliary leaf child of i in T'. Furthermore, let $\iota(v)$ be the identifier of v. Assume, for child $c_j(i)$ we already have the expression E_j. Then the multi-$(k+2)$-expression for node i, or more precisely of the third auxiliary node above it, is

$$\epsilon_{k+2}(\epsilon_{\iota(v)}(\eta_{\iota(v),k+2}(E_0 \oplus E_1 \oplus \cdots \oplus E_q))).$$

Now the following is easily proved by induction on the height of node i.

▶ **Claim 14.** *The multi-$(k+2)$-expression for node i generates the labeled graph $G_i = (V_i, E_i)$ induced by the vertices whose home is in the subtree of i. Furthermore, the set of labels of every vertex $v \in V_i$ is equal to the set of identifiers of the neighbors of v in $V \setminus V_i$.*

By the inductive hypothesis of the claim, all vertices V' in the subtree of node i that are adjacent to v in G have a label $\iota(v)$. The vertex v has a label $k+2$, but no label $\iota(v)$. Thus the operation $\eta_{\iota(v),k+2}$ creates exactly the edges between v and V'. Now, the labels $\iota(v)$ and $k+2$ can be deleted, because both have served their purpose. From every vertex labeled $\iota(v)$, the edge to v is now already constructed, and the label $k+2$ only had to mark the vertex v for the construction of these edges.

The claim for the root implies the theorem. ◀

A weaker form of Theorem 13 is the implied inequality between multi-clique-width and tree-width.

▶ **Corollary 15.** *For every graph G,* $\mathrm{mcw}(G) \leq \mathrm{tw}(G) + 2$.

As an immediate corollary, we obtain $cw(G) \leq 2^{\mathrm{tw}(G)+2}$. The tighter bound of $cw(G) \leq 2^{\mathrm{tw}(G)+1} + 1$ [12] is obtained by noticing that one could use the label $k+2$ strictly as a singleton label. Instead of deleting it with an ϵ_{k+2} operation, one could change it to the set of other labels we wanted to assign to that vertex using a $\rho_{i\to S}$ operation. The even tighter bound $cw \leq 1.5 \cdot 2^{\mathrm{tw}(G)}$ [6] is obtained by handling higher degree join nodes more efficiently. Following every binary join, the necessary edges could be inserted, allowing the number of labels to be decreased. This saves one fourth of the labels.

▶ **Corollary 16.** *There are graphs G with* $\mathrm{cw}(G) \geq 2^{\lfloor \mathrm{mcw}(G)/2 \rfloor - 2}$.

Proof. There are graphs G with $\mathrm{tw}(G) = k$ and clique-width $\mathrm{cw}(G) \geq 2^{\lfloor k/2 \rfloor - 1}$ [6]. Such graphs have multi-clique-width $\mathrm{mcw}(G) \leq k+2$ by Corollary 15. ◀

Naturally, it is easy to find graph classes with unbounded tree-width that still exhibit this exponential discrepancy between clique-width and multi-clique-width. One way is just to add a large clique, but there are many not so obvious ways.

We want to compare multi-clique-width with boolean-width.

▶ **Theorem 17.** *For every graph G,* $\mathrm{boolw}(G) \leq \mathrm{mcw}(G) \leq 2^{\mathrm{boolw}(G)}$.

Proof. boolw$(G) \leq$ mcw(G): Assuming mcw$(G) = k$, we start with a multi-k-expression for G. W.l.o.g., assume that each vertex v is created as a single vertex by the operation $m\langle i_1, \ldots, i_j\rangle$ with $m = 1$. Then, there is a bijection between the vertices V and the leaves of the parse tree T. Viewed as a graph, the other nodes of T have degrees 2 or 3. We replace all maximal paths with internal nodes of degree 2 by single edges to obtain a tree T'.

Consider any edge $e = (u, v)$ of T', where v is a descendant of u in T. Let $X \subseteq V$ be the set of vertices of the subtree T_v. For every subset $S \subseteq X$, the set $N(S) \cap \overline{X}$ of neighbors of S outside of X only depends on the union of the set of labels of the vertices of S. There are at most 2^k such subsets of labels, and thus at most 2^k such neighborhoods. The binary logarithm of the largest such number of neighborhoods over all edges of T' is an upper bound on boolw(G), i.e., boolw$(G) \leq k$.

mcw$(G) \leq 2^{\text{boolw}(G)}$: By Proposition 10, mcw$(G) \leq$ cw(G), and the inequality cw$(G) \leq 2^{\text{boolw}(G)}$ [5] is known. ◀

Even though, boolean-width has the desirable property boolw$(G) \leq$ mcw(G), sometimes more efficient algorithms are possible in terms of mcw(G) than in terms of boolw(G). Indeed, every graph property expressible in $\mathcal{MS}_1$, is decidable in linear time for graphs of bounded clique-width [10], while for arbitrary graphs of bounded boolean-width, at least quadratic time is required. Naturally, this can also be viewed as an indication of the strength of the boolean-width parameter. Even graphs without a simple structure can have small boolean-width. In the next section, we will see that for specific problems the (exponential) dependance of the running time on the multi-clique-width can be quite good.

4 Algorithms based on Multi-Clique-Width

The algorithmic purpose of clique-width and other width parameters is to put problems into FPT, i.e., making them fixed parameter tractable (see [14]). This means achieving a running time of $O(f(k)n^{O(1)})$ for an arbitrary computable function f. In reality things are not so bad. Algorithms based on clique-width often have a running time of $O(c^k n^e)$ or $O(c^{k \log k} n^e)$ with $k =$ cw(G), $n = |V|$, and c and e being small constants.

Assume that we are given a multi-k-expression for G and we have an algorithm with similar running time when k is the multi-clique-width. Then we have an exponential time speed-up when choosing the multi-clique based algorithm with running time $2^{O(k)}n^e$, instead of the clique-width based algorithm with clique-width $2^{\Omega(k)}$ and running time $2^{2^{\Omega(k)}}n^e$ for infinitely many graphs.

Indeed, we want to illustrate here that this scenario is occurring quite frequently. Let us see that there are many graphs with unbounded tree-width, multi-clique-width k, and clique-width exponential in k. Without changing the notion of multi-clique-width, we could extend the multi-k-expressions to allow arbitrary $k/2$-labeled graphs of tree-width at most $k/2 - 2$ as atoms instead of just isolated vertices. It is not had to see that such graphs could be produced by proper multi-k-expressions. $k/2$ labels are sufficient to simulate the tree-width construction, and the remaining $k/2$ labels could be placed in any possible way.

But if at least one of the tree-width $k/2-2$ components are complicated, then the clique-width is exponential in k [6]. And if the graph contains a large clique of size ℓ (which could be $\Omega(n)$), then the tree-width is at least ℓ. Thus, we have a huge class of graphs where the multi-clique-width is exponentially better than the clique-width.

Now we exhibit the ease of use of the parameter mcw(G) for Independent Set. The running time as a function of the width is roughly the same for clique-width k as for multi-clique-width k. Hence, we gain an exponential speed-up in the width parameter for all the many instances were the clique-width is exponentially bigger than the multi-clique-width.

Instead of only finding a maximum independent set, or even just computing its size, we solve the more involved problem of computing the independent set polynomial, i.e., computing the numbers of independent sets of all sizes. This is not much more difficult, and one can easily simplify the algorithm if only a maximum independent set is needed. Then the dependence of the running time on the size n goes down to linear from polynomial, while the dependence on the width k stays singly exponential. In particular, we have an FPT algorithm to compute the independent set polynomial. We refer to [11, 16] for more discussions of the fixed parameter tractability of counting problems.

It is worth mentioning that the very same problem of computing the number of independent sets of all sizes has been studied by Bui-Xuan et al. [5]. They achieve a running time of $O(\alpha^2 n^2 k 2^{2k})$ for graphs with boolean-width k and maximum independent set size α. Note that α could be as large as $\Omega(n)$. We will not elaborate on our running time, but just remark that using FFTs, $O(\alpha \log \alpha\, nk2^k)$ is sufficient for multi-clique-width k.

▶ **Definition 18.** The independent set polynomial of a graph G is

$$I(x) = \sum_{i=1}^{n} a_i x^i$$

where a_i is the number of independent sets of size i in G.

The independent set polynomial is not strong enough to describe the involvement of the different labels in the independent sets. We need to do a more detailed counting to allow recurrence equations to govern the definition of the polynomials as the labeled graph is assembled by a multi-k-expression.

▶ **Definition 19.** Let $[k] = \{1, \ldots, k\}$ be the set of vertex labels. The $[k]$-labeled independent set polynomial of a $[k]$-labeled graph G (each vertex can have multiple labels from $[k]$) is

$$P(x, x_1, \ldots, x_k) = \sum_{i=1}^{n} \sum_{(n_1,\ldots,n_k)\in\{0,1\}^k} a_{i;n_1,\ldots,n_k}\, x^i \prod_{j=1}^{k} x_j^{n_j}$$

where $n_j \in \{0, 1\}$ and $a_{i;n_1,\ldots,n_k}$ is the number of independent sets of size i in G which contain some vertices with label j if and only if $n_j = 1$.

The $[k]$-labeled independent set polynomial is used in the inductive construction. At the end, the independent set polynomial $I(x)$ is easily obtained from $[k]$-labeled independent set polynomial.

▶ **Theorem 20.** *Given a graph G with n vertices, and a multi-k-expression generating G with multi-clique-width k, the independent set polynomial $I(x)$ of G can be computed in time $O(2^k(kn)^{O(1)})$.*

Proof. Using dynamic programming, we compute the $[k]$-labeled independent set polynomial of the $[k]$-labeled graphs generated by subexpressions of the given multi-k-expression. The computation is done bottom-up in the parse tree of the given multi-k-expression.

For any atomic expression $m\langle i_1, \ldots, i_j\rangle$ creating m vertices with labels $i_1, \ldots, i_j$, we have the $[k]$-labeled independent set polynomial

$$1 + \sum_{\ell=1}^{m} \binom{m}{\ell} x^\ell x_{i_1} \cdots x_{i_j} = 1 + ((1+x)^m - 1) x_{i_1} \cdots x_{i_j}.$$

In $O(m)$ arithmetic operations, we can compute all coefficients using the recurrence $\binom{m}{\ell+1} = \binom{m}{\ell}(m-\ell)/(\ell+1)$. Thus all atomic expressions for the $n = |V|$ vertices can be computed in time $O(n)$.

If the expression E has the polynomial $\tilde{P}(x, x_1, \ldots, x_k)$, then $\eta_{i,j}(E)$ has the polynomial

$$P(x, x_1, \ldots, x_k) = \tilde{P}(x, x_1, \ldots, x_k) \mod x_i x_j,$$

i.e., terms containing x_i and x_j are deleted. This is correct, because a set of vertices is independent after the introduction of the edges between labels i and j, if and only if it was an independent set before and does not contain both labels i and j.

If the expression E has the polynomial $\tilde{P}(x, x_1, \ldots, x_k)$, then $\rho_{i\to S}(E)$ has the polynomial

$$\begin{aligned} P(x, x_1, \ldots, x_k) = \tilde{P}(x, x_1, \ldots, x_{i-1}, x_{i_1} \cdots x_{i_j}, x_{i+1}, \ldots, x_k) \\ \mod (x_{i_1}^2 - x_{i_1}) \cdots \mod (x_{i_j}^2 - x_{i_j}) \end{aligned} \tag{1}$$

for $S = \{i_1, \ldots, i_j\}$, i.e., first x_i is replaced by the product $x_{i_1} \ldots x_{i_j}$. Then squares of indeterminates are replaced by their first powers. This is correct, because we still count all independent sets. They just occur in different categories as they involve different labels.

If the expression E has the polynomial $\tilde{P}(x, x_1, \ldots, x_k)$, then $\epsilon_i(E)$ has the polynomial

$$P(x, x_1, \ldots, x_k) = \tilde{P}(x, x_1, \ldots x_{i-1}, 1, x_{i+1}, \ldots, x_k),$$

i.e., the indeterminate x_i is replaced by 1. This is correct, because it is just a special case $\rho_{i\to S}(E)$.

If the expression E_ℓ ($\ell \in \{1, 2\}$) has the polynomial $\tilde{P}_\ell(x, x_1, \ldots, x_k)$, then the expression $E_1 \oplus E_2$ has the polynomial

$$\begin{aligned} P(x, x_1, \ldots, x_k) = \tilde{P}_1(x, x_1, \ldots, x_k)\, \tilde{P}_2(x, x_1, \ldots, x_k) \\ \mod (x_1^2 - x_1) \cdots \mod (x_k^2 - x_k), \end{aligned} \tag{2}$$

i.e., in the product of the polynomials, every x_i^2 is replaced by x_i, as we only care about the occurrence of a label and not about the multiplicity of such an occurrence. This is correct, because every independent set of G_1 can be combined with every independent set of G_2 to form an independent set of the join graph G, and every independent set of G can be formed in this way.

Finally,

$$I(x) = \sum_{i=1}^{n} a_i x^i = \sum_{i=1}^{n} \sum_{(n_1,\ldots,n_k)\in\{0,1\}^k} a_{i,n_1,\ldots,n_k} x^i = P(x, 1, \ldots, 1),$$

because
$a_i = \sum_{(n_1,\ldots,n_k)\in\{0,1\}^k} a_{i,n_1,\ldots,n_k}$.

To bound the running time, one should notice that the polynomial $P(x, x_1, \ldots, x_k)$ has $2^k(n+1)$ coefficients.

The polynomial $\tilde{P}(x, x_1, \ldots, x_{i-1}, x_{i_1} \cdots x_{i_j}, x_{i+1}, \ldots, x_k)$ in Eq. (1) has at most $2^k(n+1)$ non-zero coefficients, not $3^k(n+1)$, because the substitution does not increase the number of monomials. If the product in Eq. (2) is computed by school multiplication, then the running time is $O(4^k(kn)^{O(1)})$. But with a fast Fourier transform (evaluating the polynomial for $x_i = 0$ and $x_1 = 1$ for all i), the time is only $O(2^k(kn)^{O(1)})$. ◀

The easier problem of just finding the size of a maximum independent set (rather than computing the numbers of independent sets of all sizes) is now trivial. At each stage, for all exponents $n_1, \ldots, n_k$, the coefficient $a_{i;n_1,\ldots n_k}$ is only stored for the largest i with $a_{i;n_1,\ldots,n_k} \neq 0$.

▶ **Corollary 21.** *A maximum independent set can be found in time $O(2^k k^{O(1)} n)$ in graphs with multi-clique-width k.*

Proof. If during the dynamic programming algorithm to compute the size of a maximum independent set, one always stores where the largest exponent i came from, then at the end, one can easily backtrace to actually find a maximum independent set. ◀

As an additional example, we consider the NP-complete decision problem c-coloring, asking whether the input graph G can be colored with c colors for a constant integer $c \geq 3$, such that no adjacent vertices have the same color.

▶ **Theorem 22.** *For graphs G of multi-clique-width k with a given multi-k-expression for G, and any positive integer constant c, the c-coloring problem can be solved in time $2^{O(ck)} n$.*

Proof. We present a dynamic programming algorithm based on the parse tree structure of the multi-k-expression. We classify the colorings of the graphs generated by sub-expressions according to the labels used for the vertices of each color. Let Q with $|Q| = c$ be the set of colors and L with $|L| = k$ be the set of labels. Let $B_1, \ldots, B_r$ with $r = 2^{ck}$ be the sequence (say in lexicographic order) of all bipartite graphs with the left vertex set Q and the right vertex set L. Let E_p be the set of edges in B_p. For every subexpression F, we define $F(B_p)$ to be true, if and only if the following holds. The graph generated by F can be colored with Q such that some vertex colored with $q \in Q$ is labeled with a set of labels containing $i \in L$, if and only if (q, i) is an edge in B_p.

We now show that $F(B_p)$ can easily be computed from all the $F'(B_{p'})$ where F' is a subexpression of F and $p' \in \{1, \ldots, r\}$. We analyze according to the structure of F.

If F is an atomic expression $m\langle i_1, \ldots, i_j\rangle$ creating m vertices with labels $i_1, \ldots, i_j$, then $F(B_p)$ is true, if and only if $E_p = \{(q, i) \mid q \in Q' \text{ and } i \in \{i_1, \ldots, i_j\}\}$ for some Q' with $\emptyset \neq Q' \subseteq Q$.

If $F = \eta_{i,j}(F')$, then $F(B_p)$ is true, if and only if $F'(B_p)$ is true and for no color $q \in Q$ there are both edges (q, i) and (q, j) present in B_p. In other words, a previous coloring is still valid, if and only if no color appears at both endpoints of newly added edges.

If $F = \rho_{i\to S}(F')$, then $F(B_p)$ is true, if and only if $F'(B_{p'})$ is true for some p' with

$$E_p = \{(q, \ell) \mid (q, \ell) \in E_{p'} \text{ and } \ell \neq i\} \cup \{(q, j) \mid (q, i) \in E_{p'} \text{ and } j \in S\}.$$

If $F = F' \oplus F''$, then $F(B_p)$ is true, if and only if $F'(B_{p'})$ is true and $F''(B_{p''})$ is true for some p', p'' with $E_p = E_{p'} \cup E_{p''}$.

Given this simple characterization of $F(B_p)$ in terms of $F'(B_{p'})$ for some p' and the immediate sub-expressions F', it should be immediately clear how the value of $F(B_p)$ can be computed, when the values of the $F'(B_{p'})$ are known.

Furthermore, it is a simple proof by induction on the structure of an expression F that $F(B_p)$ is true, if and only if the graph generated by F can be colored with Q such that some vertex colored with $q \in Q$ is labeled with a set of labels containing $i \in L$, if and only if (q, i) is an edge in B_p.

Naturally, at the end, the graph generated by F is k-colorable, if and only if $F(B_p)$ is true for some B_p.

The running time is linear in n, because there are $O(n)$ nodes to process and the time spent in every node only depends on the number c of colors and the number k of labels. In every node, an array of 2^{ck} boolean values (one for each bipartite graph on the vertex sets Q and L) has to be processed in a simple fashion. The resulting running time is $2^{O(ck)}n$.

There is quite some waste of time involved in handling all the bipartite graphs on the vertex sets Q and L, because the truth value for a graph B_j does not change, when the set of colors Q and the set of labels L are permuted in an arbitrary way. This does not mean that the running time can be divided by $c!k!$, because typically many such permutations are automorphisms not creating new bipartite graphs. The exact number of isomorphism types of such bipartite graphs can be computed with the Redfield-Pólya enumeration theorem (see [23]), but that does not result in a nicer upper bound. Clearly, any practical implementation would do the computation for just one bipartite graph for every isomorphism type. ◀

5 Conclusions and Open Problems

We have proposed a powerful parameter multi-clique-width. It allows us to achieve faster running times for natural classes of graphs and interesting algorithmic tasks. Assume, we are given the input graph by a multi-k-expression. Then we have very efficient algorithms for this class of graphs, as illustrated by the independent set polynomial and the coloring problem. On the other hand, for any algorithm based on clique-width, we could only get exponentially slower (in k) algorithms for the same problems and the same collection of graphs. Also, equally efficient algorithms are not known based on rank-width or boolean-width, when the corresponding decompositions are given.

Most questions related to the new multi-clique-width are still open. Is it difficult to compute or approximate? We expect it to be NP-hard, like clique-width. We also conjecture it to be in FPT (fixed parameter tractable) and to be constant factor approximable in time singly exponential in the multi-clique-width and linear in the length like tree-width. But obviously this is very difficult, as it is also open for clique-width.

A main question is whether most algorithms for clique-width k, can be extended to work with similar efficiency for multi-clique-width k. We have illustrated that this is the case for some interesting counting and decision problems. On the other hand, there is the question of identifying the problems where this is not the case.

References

1 Stefan Arnborg, D. G. Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM Journal of Alg. and Discrete Methods*, 8:277–284, 1987.

2 Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, 1996. doi:10.1137/S0097539793251219.

3 Hans L. Bodlaender, Pål G. Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michal Pilipczuk. An $O(c^k n)$ 5-approximation algorithm for treewidth. In *Proc. 54th FOCS 2013*, pages 499–508. IEEE, 2013.

4 Hans L. Bodlaender, John R. Gilbert, Hjálmtyr Hafsteinsson, and Ton Kloks. Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. *J. Algorithms*, 18(2):238–255, 1995. doi:10.1006/jagm.1995.1009.

5 Binh-Minh Bui-Xuan, Jan Arne Telle, and Martin Vatshelle. Boolean-width of graphs. *Theor. Comput. Sci.*, 412(39):5187–5204, 2011. doi:10.1016/j.tcs.2011.05.022.

6 Derek G. Corneil and Udi Rotics. On the relationship between clique-width and treewidth. *SIAM J. Comput.*, 34(4):825–847, 2005. doi:10.1137/S0097539701385351.

7 Bruno Courcelle. The monadic second-order logic of graphs. I. recognizable sets of finite graphs. *Inf. Comput*, 85(1):12–75, March 1990. doi:10.1016/0890-5401(90)90043-H.

8 Bruno Courcelle, Joost Engelfriet, and Grzegorz Rozenberg. Handle-rewriting hypergraph grammars. *J. Comput. Syst. Sci.*, 46(2):218–270, 1993. doi:10.1016/0022-0000(93)90004-G.

9 Bruno Courcelle and Johann A. Makowsky. Fusion in relational structures and the verification of monadic second-order properties. *Mathematical Structures in Computer Science*, 12(2):203–235, 2002.

10 Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000. doi:10.1007/s002249910009.

11 Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. *Discrete Applied Mathematics*, 108(1-2):23–52, 2001. doi:10.1016/S0166-218X(00)00221-3.

12 Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000. doi:10.1016/S0166-218X(99)00184-5.

13 Bruno Courcelle and Andrew Twigg. Constrained-path labellings on graphs of bounded clique-width. *Theory Comput. Syst.*, 47(2):531–567, 2010. URL: http://springerlink.metapress.com/content/b3268gtk313180q0/.

14 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

15 Michael R. Fellows, Frances A. Rosamond, Udi Rotics, and Stefan Szeider. Clique-width minimization is NP-hard. In Jon M. Kleinberg, editor, *STOC*, pages 354–362. ACM, 2006. doi:10.1145/1132516.1132568.

16 Eldar Fischer, Johann A. Makowsky, and Elena V. Ravve. Counting truth assignments of formulas of bounded tree-width or clique-width. *Discrete Applied Mathematics*, 156(4):511–529, 2008. doi:10.1016/j.dam.2006.06.020.

17 Fedor V. Fomin, Daniel Lokshtanov, Michal Pilipczuk, Saket Saurabh, and Marcin Wrochna. Fully polynomial-time parameterized computations for graphs and matrices of low treewidth. *CoRR*, abs/1511.01379, 2015. URL: http://arxiv.org/abs/1511.01379.

18 Martin Fürer. A natural generalization of bounded tree-width and bounded clique-width. *Proceedings of LATIN 2014: Theoretical Informatics - 11th Latin American Symposium. Springer LNCS*, 8392:72–83, 2014. doi:10.1007/978-3-642-54423-1.

19 Robert Ganian and Petr Hlinený. On parse trees and myhill-nerode-type tools for handling graphs of bounded rank-width. *Discrete Applied Mathematics*, 158(7):851–867, 2010. doi:10.1016/j.dam.2009.10.018.

20 Petr Hlinený and Sang il Oum. Finding branch-decompositions and rank-decompositions. *SIAM J. Comput*, 38(3):1012–1032, 2008. doi:10.1137/070685920.

21 Sang-il Oum. Approximating rank-width and clique-width quickly. *ACM Trans. Algorithms*, 5(1):10:1–10:20, December 2008. doi:10.1145/1435375.1435385.

22 Sang-il Oum and Paul D. Seymour. Approximating clique-width and branch-width. *J. Comb. Theory, Ser. B*, 96(4):514–528, 2006. doi:10.1016/j.jctb.2005.10.006.

23 G. Pólya and R. C. Read. *Combinatorial Enumeration of Groups, Graphs and Chemical Compounds*. Springer-Verlag, New York, 1987.

24 Neil Robertson and Paul D. Seymour. Graph minors. III. Planar tree-width. *J. Comb. Theory, Ser. B*, 36(1):49–64, 1984. doi:10.1016/0095-8956(84)90013-3.

25 Sigve Hortemo Sæther and Jan Arne Telle. Between treewidth and clique-width. In Dieter Kratsch and Ioan Todinca, editors, *Graph-Theoretic Concepts in Computer Science - 40th International Workshop, WG 2014, Nouan-le-Fuzelier, France, June 25-27, 2014. Revised Selected Papers*, volume 8747 of *Lecture Notes in Computer Science*, pages 396–407. Springer, 2014. doi:10.1007/978-3-319-12340-0.

Computational Tradeoffs in Biological Neural Networks: Self-Stabilizing Winner-Take-All Networks*

Nancy Lynch[1], Cameron Musco[2], and Merav Parter[3]

1 **Massachusetts Institute of Technology, Cambridge, USA**
`lynch@csail.mit.edu`
2 **Massachusetts Institute of Technology, Cambridge, USA**
`cnmusco@mit.edu`
3 **Massachusetts Institute of Technology, Cambridge, USA**
`parter@mit.edu`

Abstract

We initiate a line of investigation into biological neural networks from an algorithmic perspective. We develop a simplified but biologically plausible model for distributed computation in *stochastic spiking neural networks* and study tradeoffs between computation time and network complexity in this model. Our aim is to abstract real neural networks in a way that, while not capturing all interesting features, preserves high-level behavior and allows us to make biologically relevant conclusions.

In this paper, we focus on the important 'winner-take-all' (WTA) problem, which is analogous to a neural leader election unit: a network consisting of n input neurons and n corresponding output neurons must converge to a state in which a single output corresponding to a firing input (the 'winner') fires, while all other outputs remain silent. Neural circuits for WTA rely on inhibitory neurons, which suppress the activity of competing outputs and drive the network towards a converged state with a single firing winner. We attempt to understand how the number of inhibitors used affects network convergence time.

We show that it is possible to significantly outperform naive WTA constructions through a more refined use of inhibition, solving the problem in $O(\theta)$ rounds in expectation with just $O(\log^{1/\theta} n)$ inhibitors for any θ. An alternative construction gives convergence in $O(\log^{1/\theta} n)$ rounds with $O(\theta)$ inhibitors. We complement these upper bounds with our main technical contribution, a nearly matching lower bound for networks using $\geq \log\log n$ inhibitors. Our lower bound uses familiar indistinguishability and locality arguments from distributed computing theory applied to the neural setting. It lets us derive a number of interesting conclusions about the structure of any network solving WTA with good probability, and the use of randomness and inhibition within such a network.

1998 ACM Subject Classification F.1.1 Models of Computation – Unbounded-action devices, C.1.3 Other Architecture Styles – Neural nets

Keywords and phrases biological distributed algorithms, neural networks, distributed lower bounds, winner-take-all networks

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.15

* This work was partially supported by NSF Graduate Research Fellowship No. 1122374, AFOSR grant FA9550-13-1-0042 and the NSF Center for Science of Information.

© Nancy Lynch, Cameron Musco, and Merav Parter;
licensed under Creative Commons License CC-BY
8th Innovations in Theoretical Computer Science Conference (ITCS 2017).
Editor: Christos H. Papadimitrou; Article No. 15; pp. 15:1–15:44

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

In this paper, we study biological neural networks from an algorithmic perspective, focusing on understanding tradeoffs between computation time and network complexity. We use a biologically plausible yet simplified neural computational model. Our goal is to abstract real neural networks in a way that, while not capturing all interesting features, preserves high-level behavior and allows us to make biologically relevant conclusions.

1.1 Model and Problem Statement

Model

We work with *spiking neural networks* (SNNs) [26, 27, 12, 18, 15], in which neurons fire in discrete pulses, in response to a sufficiently high membrane potential. This potential is induced by spikes from neighboring neurons, which can have either an excitatory or inhibitory effect (increasing or decreasing the potential). Our model is *stochastic* – each neuron functions as a probabilistic threshold unit, spiking with probability given by applying a sigmoid function to its membrane potential. In this respect, our networks are similar to the popular Boltzmann machine [1], with the important distinction that synaptic weights are not required to be symmetric and, as observed in nature, neurons are either strictly inhibitory (all outgoing edge weights are negative) or excitatory. While a rich literature focuses on deterministic threshold circuits [31, 16] we employ a stochastic model as it is widely accepted that neural computation is inherently stochastic [3, 42, 10], and that while this can lead to a number of challenges, it also affords significant computational advantages [30].

The WTA Problem

We focus on the Winner-Take-All (WTA) problem, which is one of the most studied problems in computational neuroscience. A WTA network has n input neurons, n corresponding outputs, and a set of auxiliary neurons that facilitate computation. The goal is to pick a 'winning' input – that is, the network should produce a single firing output which corresponds to a firing input. Often the winning input is the one with the highest firing rate, in which case WTA serves as a neural max function. We focus on the case when all inputs have the same or similar firing rates, in which case WTA serves as a leader election unit.

WTA is widely applicable, including in circuits that implement visual attention via WTA competition between groups of neurons that process different input classes [21, 23, 17]. It is also the foundation of competitive learning [32, 20, 14], in which classifiers compete to respond to specific input types. More broadly, WTA is known to be a powerful computational primitive [28, 29] – a network equipped with WTA units can perform some tasks significantly more efficiently than with just linear threshold neurons (McCulloch-Pitts neurons or perceptrons).

Related Work

Due to its importance, there has been significant work on WTA, including in biologically plausible spiking networks [22, 48, 43, 7, 47, 34, 33, 2]. This work is extremely diverse – while mathematical analysis is typically given, different papers show different guarantees and apply varying levels of rigor. To the best of our knowledge, no asymptotic time bounds (e.g., as a function of the number of inputs n) for solving WTA in spiking neural networks have been

established.[1] Additionally, previous analysis often requires a specific initial network state to show convergence and does not show that the network is self-stabilizing and converges from an arbitrary starting state, as is necessary in a biological system.

Within theoretical computer science, our work is most inspired by: (1) work on the computational power of spiking neural networks, including the power of WTA as a black-box primitive, most notably by Maass et al. [27, 28, 29] (2) the pioneering work of Les Valiant on the neuroidal model [44, 45, 46] and (3) self-stabilization algorithms in distributed networks [8, 25]. We survey this literature in more depth in Appendix A.1.

Basic WTA Networks

We restrict our attention to a simple network structure that can implement WTA efficiently using a small number of auxiliary neurons. A network consists of three layers: n input neurons X, n output neurons Y, and α auxiliary neurons Z. We usually assume all auxiliary neurons are inhibitory, however in Appendix C give extensions to the more general case where we allow auxiliary neurons to also be excitatory. Similar to well-known feedforward networks, all synaptic connections are between layers[2] with the exception of an excitatory self-loop from each output y_i to itself. This basic structure is biologically plausible; in particular self-loops and reciprocal excitatory-inhibitory connections (as implemented in our networks) are used in many biological models of WTA computation [48, 7, 38].

It is well known that inhibition is crucial for solving WTA – outputs compete for activation via *lateral inhibition* or *recurrent inhibition* [7, 38]. In our network, outputs fire in response to stimulation by their corresponding inputs, thereby stimulating inhibitors which suppress the activity of other outputs. Once a single winner is selected, it must remain distinguished from the remainder of the outputs. This is achieved via positive feedback – a consistently firing output will tend to continue firing due to its excitatory self-loop.

1.2 Our Contribution

Computational Tradeoffs

We explore the tradeoff between the number of inhibitors α used in a WTA network (i.e., the complexity of the network) and the time required to select a winning output (to converge to a WTA state). In artificial neural networks, inhibitory and excitatory connections are often treated equally, as connections with either positive or negative weights. However, in reality, neurons themselves are either inhibitory or excitatory and do not have outgoing connections of both types. There are many fewer inhibitors (around 15% of the neural population [39, 13]), and they typically have restricted connectivity structures, often inhibiting just neurons in their local vicinity [29]. This gives natural motivation to understanding how the number of inhibitors used in a network affects its computational power. We give two main results:

▶ **Theorem 1** (Upper bound). *(1) For any $\alpha \geq 2$ there exists a basic WTA network with α inhibitors that, from any* arbitrary *starting configuration, converges to a valid WTA state in $O(\alpha \log^{1/\alpha} n)$ expected time. (2) For any $\theta \geq 1$ there exists a basic WTA network with $\alpha = O(\theta \log^{1/\theta} n)$ inhibitors that converges in $O(\theta)$ expected time.*

[1] Aside from immediate bounds for deterministic circuits using many ($\Omega(n)$) auxiliary neurons [22, 29].
[2] Although, due to recurrent connections the network convergence time is not synonymous with the number of layers.

For $\alpha \geq \log\log n$ the above gives runtime $\tilde{O}\left(\frac{\log\log n}{\log \alpha}\right)$. We give a near matching lower bound in this case, which holds even if we allow both excitatory and inhibitory auxiliary neurons.

▶ **Theorem 2** (Lower bound). *Any basic WTA network with α inhibitors requires $\Omega(\log\log n/\log\alpha)$ rounds to solve WTA in expectation.*

Upper Bound Techniques

Our upper bounds are based on random competition between outputs that fire in response to stimulation from their firing inputs. One "stability" inhibitor is responsible for maintaining a WTA steady-state: as soon as just a single output fires in a round it becomes the winner of the network. Its positive feedback self-loop allows it to keep firing in subsequent rounds, while all other outputs do not fire due to inhibition from the stability inhibitor.

In order to reach a round in which just a single output fires, we employ a number of "convergence inhibitors". Ideally, if k competing outputs fire in a round, each would fire in the next round with probability $1/k$ and we would have just a single firing output with constant probability. We can approximate this behavior using $\lfloor \log n \rfloor$ convergence inhibitors, each of which acts as a threshold circuit and fires whenever $\geq 2^i$ outputs fire for $i \in 1, ..., \lfloor \log n \rfloor$. Thus when k outputs fire, approximately $\log(k)$ inhibitors fire, the inhibition causes outputs to continue firing with probability $\Theta(1/k)$, and convergence is achieved in constant rounds in expectation. This technique implicitly splits the possible number of firing outputs into $\log n$ *density classes* and uses one inhibitor to ensure fast convergence from each class. To obtain more general runtime tradeoffs, we will use density classes of increasing coarseness, with the inhibitors assigned to each density classes ensuring that the number of firing outputs decreases in few rounds until it falls into a finer density class, and eventually until just a single output fires.

Lower Bound Techniques

Our lower bound shows that *any* network which solves WTA must have a similar structure to the network described above. The inhibitory neurons can always be roughly be divided into two classes: stability and convergence inhibitors. Further, while randomness is important in breaking symmetry between competing inputs, we show that in any efficient network, the inhibitors behave in a *nearly deterministic* manner, matching behavior seen in our upper bounds. After significantly constraining inhibitor behavior, we are able to analyze how any network which solves WTA behaves on inputs with varying numbers of firing neurons. Specifically, we consider $\Theta(\log n)$ different inputs configurations, with geometrically increasing numbers of firing input neurons, ranging from $O(1)$ to $O(n)$. We show that, after t rounds, with good probability, the network *does not distinguish between* (i.e. behaves identically for) $\Theta(\log n/\alpha^t)$ inputs.

As long as $\log n/\alpha^t > 2$, after t rounds, there are at least two inputs not distinguished by the network, and so on which the network cannot achieve WTA with good probability. This yields our lower bound of $t = \Omega(\log\log n/\log\alpha)$ rounds in expectation. Our argument uses techniques familiar in distributed computing theory [24], showing that limited local information prevents outputs from behaving in distinct manners for a large number of density classes in each round.

We obtain a corresponding lower bound for the number of rounds required to solve WTA *with high probability* by showing that in general, the high probability runtime is $\Omega(\log n/\log\log\log n)$ times the expected runtime. This nearly matches the $O(\log n)$ gap which can be achieved by noting that in $O(\log n)$ runs, any network will converge within its

Table 1 Expected Time vs. Number of Inhibitors Tradeoff in Basic WTA Networks.

Inhibitors	Lower Bound (Expected Time)	Upper Bound (Expected Time)
Unbounded	$\Omega(1)$ ($\Omega(\log n)$ high probability time)	$O(1)$ with $\alpha = \Theta(\log^{1/c} n)$
1	$\Omega(n^c)$	$O(n^c)$
2	$\Omega(\log n / \log\log n)$	$O(\log n)$
α	$\Omega(\log\log n / \log \alpha)$	$O(\alpha \cdot \log^{1/\alpha} n)$, for $\alpha = O(\log\log n)$ $\tilde{O}\left(\frac{\log\log n}{\log \alpha}\right)$, for $\alpha = \Omega(\log\log n)$

expected runtime at least once with high probability. Our conversion result shows that, in our setting, expected runtime is a more natural metric – it is controlled by the number of inhibitors used, whereas the high probability runtime is just a function of expected runtime, independent of the number of inhibitors

1.3 Biological Insights in Our Results

Previous work has conjectured that widespread use of simple WTA implementations in the brain may explain how complex computation is possible even when inhibition is relatively limited and localized [29]. Our work shows that WTA can be achieved and maintained efficiently using very few inhibitors and with a very simple connectivity structure.

Our upper and lower bound constructions have a common take home message that may shed some light into the biological implementations of WTA networks. For instance, the division of inhibitors into "task preservers" (stability inhibitors) and "task solvers" (convergence inhibitors) seems fundamental. Further, while randomness is crucial as it allows for symmetry breaking amongst competing outputs, it appears (both in the upper bounds and the corresponding lower bound) that in optimal networks the inhibitors behave almost as deterministic threshold circuits, firing with high probability whenever the number of firing outputs is above a certain level. This presents an interesting dichotomy – while randomness is necessary computationally, it also has a cost in leading to unpredictable behavior amongst the inhibitors which 'control' the network.

1.4 Road Map

In Sec. 2 we describe our spiking neural network model and specify the WTA problem. In Sec. 3 we give two warm up examples of WTA networks to illustrate the tradeoff between convergence time and network size. The first has two inhibitors and converges to the WTA state within $O(\log n)$ rounds in expectation. The second has $O(\log n)$ inhibitors and $O(1)$ expected runtime. In Sec. 4.1, we provide more delicate constructions for any number of inhibitors α. Our key technical result appears in Sec. 4.2 where we provide a runtime lower bound (both for expected and high probability time) for circuits using α inhibitors, for any α. Our lower bound nearly matches our upper bounds for $\alpha = \Omega(\log\log n)$. Missing proofs are deferred to the Appendix.

2 Neural Network Model

A *Spiking Neural Network* (SNN) $\mathrm{N} = \langle \mathrm{X}, \mathrm{Y}, \mathrm{Z}, w, b \rangle$ consists of n input neurons $\mathrm{X} = \{x_1, \ldots, x_n\}$, n output neurons $\mathrm{Y} = \{y_1, \ldots, y_n\}$, and α auxiliary neurons $\mathrm{Z} = \{z_1, ..., z_\alpha\}$. The directed, weighted synaptic connections between X, Y, and Z are described by the weight function $w : [\mathrm{X} \cup \mathrm{Y} \cup \mathrm{Z}] \times [\mathrm{X} \cup \mathrm{Y} \cup \mathrm{Z}] \to \mathbb{R}$. The in-degree of every input neuron x_i is zero.

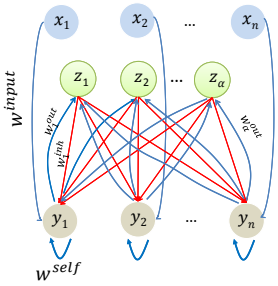


Figure 1 Basic WTA Network structure.

Each neuron is either inhibitory or excitatory: if v is inhibitory, then $w(v,u) \leq 0$ for every u, and if v is excitatory, then $w(v,u) \geq 0$ for every u. Finally, for any neuron v, $b(v) \in \mathbb{R}_{\geq 0}$ is the activation bias – as we will see, roughly, v's membrane potential must reach $b(v)$ for a spike to occur with good probability.

The Basic WTA Network and its Dynamics:

We focus on a restricted class of *basic SNNs*, in which all auxiliary neurons are inhibitory, inputs connect only to their corresponding outputs, and there are no connections within the inhibitory or output layers, aside from an excitatory self-loop from each output to itself. All outputs have identical parameters, i.e., bias values and edge weights.

We introduce some more concise notation to describe basic SNNs. Let $\mathrm{w}^{\mathrm{input}} > 0$ be the synaptic weight from each input x_j to its corresponding output y_j. Let $\mathrm{w}^{\mathrm{self}} > 0$ be the weight of the excitatory self-loop from output y_j to itself. Let $\mathrm{w}^{\mathrm{inh}}{}_j \leq 0$ be the weight of each inhibitory synapse from inhibitor z_j to an output neuron. Conversely, let $\mathrm{w}^{\mathrm{out}}{}_j \geq 0$ be the weight of each excitatory synapse from an output in Y to inhibitor z_j. Finally, let $\mathrm{b}^{\mathrm{out}}$ be the bias value for each output neuron. For an diagram of the basic architecture, see Fig. 1.

The network evolves in discrete, synchronous *rounds* as a Markov chain, with an alternating dynamic between the neurons in X, Y and Z. We give in-depth biological motivation in Appendix A.2. Each round t consists of three sub-rounds denoted by $(t,1),(t,2)$ and $(t,3)$ where the three layers inputs, outputs and inhibitors are scheduled to fire: In the first sub-round $(t,1)$ of each round t, the input layer fires. We consider static inputs so each x_i either fires in every round or does not fire in any round. After that, in sub-round $(t,2)$ the output neurons in Y spike with probabilities dependent on their membrane potentials. Finally, in sub-round $(t,3)$ the inhibitors in Z spike in response to their potentials. The firing probability of every neuron depends on the firing status of its neighboring neurons in the preceding three sub-rounds (i.e., a length of one round). This probabilistic firing is modeled using a standard sigmoid function. For each neuron u, and each round $t \geq 1$, let $u^{(t,k)} = 1$ if u fires (i.e., generates a spike) in sub-round (t,k) for $k \in \{1,2,3\}$. Let $u^{0,k}$ denote the initial firing state of the neuron – we will discuss how this is determined below.

Since each neuron is always scheduled to fire in one of $(t,1)$, $(t,2)$ or $(t,3)$ depending on whether it is in layer X, Y, or Z, for convenience we will often omit the sub-round notation, writing $u^t = 1$ if u fires in *one* of the sub-rounds (t,k). We call u^t, the *firing state* of u in round t. Informally, we say that u *fires in round* t if $u^t = 1$. For each output $y_j \in \mathrm{Y}$ and every $t \geq 1$, let $pot(y_j,t)$ denote the membrane potential at sub-round $(t,2)$ and $p(y_j,t)$

denote the corresponding firing probability. These values are calculated as:

$$pot(y_j,t) = (x_j^{(t,1)}\, \mathrm{w}^{\mathrm{input}}) + (y_j^{(t-1,2)} \cdot \mathrm{w}^{\mathrm{self}}) + \left[\sum_{z_i \in \mathrm{Z}} z_i^{(t-1,3)} \cdot \mathrm{w}^{\mathrm{inh}}{}_i\right] - \mathrm{b}^{\mathrm{out}}$$
$$\text{and } p(y_j,t) = \frac{1}{1+e^{-pot(y_j,t)/\lambda}} \tag{1}$$

where $\lambda > 0$ is a *temperature parameter*, which determines the steepness of the sigmoid. Note that (1) incorporates excitatory and inhibitory effects from any spikes occurring within the three sub-rounds before the outputs spike in sub-round $(t,2)$. Specifically, this includes input spikes in sub-round $(t,1)$ along with output and inhibitory spikes in sub-rounds $(t-1,2),(t-1,3)$ respectively. Also note that when $t=1$, the firing probability depends on the initial firing states $x_j^{(0,1)}$, $y_j^{(0,2)}$ and $z_i^{(0,3)}$. We will discuss how these are determined below. Applying the same rules, in sub-round $(t,3)$, each inhibitor in Z fires with probability $p(z_j,t)$ calculated as:

$$pot(z_j,t) = \left[\sum_{y_i \in \mathrm{Y}} y_i^{(t,2)} \cdot \mathrm{w}^{\mathrm{out}}{}_j\right] - b(z_j) \text{ and } p(z_j,t) = \frac{1}{1+e^{-pot(z_j,t)/\lambda}}. \tag{2}$$

Again (2) incorporates effects from relevant spikes within three sub-rounds $(t-1,3),(t,1)$ and $(t,2)$. However, since inhibitors are connected only to outputs, the only sub-round that affects them is $(t,2)$. After the inhibitors fire, we proceed to round $t+1$, beginning with the firing of the inputs.

We finally specify how the initial firing states are determined. As inputs are static, $x_j^{(0,1)}$ is 1 for firing inputs and 0 for non-firing inputs. $y_j^{(0,2)}$ is arbitrary, while $z_i^{(0,3)}$ is determined as in any regular round according to (2) below with $t=0$ (and so depends on each $y_j^{(0,2)}$). It is not hard to see that is equivalent to just allowing all initial firing states to be arbitrary. This would lead to arbitrary $y_i^{(1,2)}$ and $z_i^{(1,3)}$ determined according to equation (2), which matches our model if we relabel the states in round 1 to be the initial states.

Temperature and Background Noise

It is clear that the temperature λ does not affect the computational power of the network as we can simply adjust all synapse weights and neuron biases by a factor of λ/λ' to simulate a network with temperature λ'. Hence, we can fix λ to make exposition easier. In our proofs we will always set $\lambda = 1/\Theta(\log n)$. We assume that neurons in Z, Y have bias $b(v) = \Omega(\lambda \log n)$, so they do not fire with probability $1 - 1/(1+e^{-c\cdot\log n}) = 1 - 1/n^c$ when they receive no external stimulation. We call this the *no-background noise* assumption: the network is quiet when no input is introduced. This assumption is used only for technical reasons in our general α inhibitor lower bound. We are hopeful that it could be removed.

System Configuration

For any $t \geq 1$, the configuration $\mathcal{C}^t = (\mathrm{X}^t, \mathrm{Y}^t, \mathrm{Z}^t)$ in round t is defined by the firing states[3] of the corresponding neurons in round t where $\mathrm{X}^t = [x_1^t, ..., x_n^t]$ and Y^t and Z^t are defined analogously. Recall that $x_i^t = 1, y_i^t = 1, z_i^t = 1$ if the input x_i (output y_i, inhibitor z_i) fires in

[3] The firing state of a neuron is a binary number indicating if it is firing or not.

sub-round $(t, 1)$ (resp., $(t, 2), (t, 3)$). We consider a static input setting where $\mathrm{X}^t = \mathrm{X}$ for all t.[4] We abuse notation slightly, thinking of X as a vector of binary input values where $x_j = 1$ indicates that x_j fires in every round ($x_j^t = 1$ for all t) and $x_j = 0$ implies that x_j never fires ($x_j^t = 0$ for all t). We denote the the initial configuration $\mathcal{C}^0$. As discussed we have $\mathrm{X}^0 = \mathrm{X}$, Y^0 arbitrary, and Z^0 determined as in any round according to equation (2).

The WTA Problem

A binary winner-take-all network given n inputs should converge to having a single firing output corresponding to a firing input (the 'winner'), if one exists. Formally, given $\mathrm{X} \in \{0,1\}^n$, let $f(\mathrm{X}) = \{\mathrm{Y} \in \{0,1\}^n \mid y_i \leq x_i \ \forall i \text{ and } \| \mathrm{Y} \|_1 = \min(1, \| \mathrm{X} \|_1)\}$ where $\| \cdot \|_1$ is the standard 1-norm, used to denote the number of firing neurons in a set.

We say N *satisfies WTA* in round t if $\mathrm{Y}^t \in f(\mathrm{X})$. We say N *converges to WTA* in t rounds with probability $1 - \delta$ if for every input $\mathrm{X} \in \{0,1\}^n$ and every initial output configuration Y^0, with probability at least $1 - \delta$, $\mathrm{Y}^t \in f(x)$ and $\mathrm{Y}^{t'} = \mathrm{Y}^t$ for all $t' \in [t+1, t+n^c]$ where c is a positive constant.[5] That is, the network satisfies WTA in round t and maintains the satisfying configuration for polynomial in n subsequent rounds. As our neurons are inherently probabilistic, our definition of convergence is as well – we will never be able to avoid occasional random deviations from a correct output state and so just demand that the state is maintained a large number of rounds.

We let $\mathcal{ET}(\mathrm{N})$ denote the maximum expected time required to converge to WTA, taken over all possible inputs X and initial output configurations Y^0. In the same manner, $\mathcal{HT}(\mathrm{N})$ denotes the maximum time required for convergence to WTA with high probability.[6]

3 Warm Up: Two Simple Networks for WTA

We begin by presenting two WTA networks that represent two extremes of the inhibitor-time tradeoff. They also illustrate the rough intuition that will appear in our later network constructions and lower bound strategies.

WTA with Two Inhibitors

In our two inhibitor network we have $\mathrm{Z} = \{z_s, z_c\}$. The neuron z_s is a *stability* inhibitor that maintains the WTA state once it has been reached. It fires w.h.p. in sub-round $(t, 3)$ whenever at least one output fires in sub-round $(t, 2)$. The neuron z_c is a *convergence* inhibitor that fires w.h.p. whenever WTA has not yet been reached – i.e. whenever ≥ 2 outputs fire in sub-round $(t, 2)$.

We set the weights connecting z_s and z_c to the outputs such that when both fire in round t, any output that fired in round t will fire with probability $1/2$ in round $t+1$. Any output that *did not fire* in round t will not fire in round $t+1$ w.h.p. as it will not have an active excitatory self-loop and so its membrane potential will be too low to overcome the inhibition.

---

[4] Note however that our model can easily handle non-static inputs. All algorithms given will converge from an arbitrary initial configuration and so will converge if X changes.

[5] Formally, a family if networks $\mathcal{N} = \{N(n)\}$ for all integers $n \geq 1$ converges to WTA in $t(n)$ rounds with probability $1 - \delta$ if there exists $c > 0$ such that for all n, for all $X \in \{0,1\}^n$ and for all $Y^0 \in \{0,1\}^n$, with probability at least $1 - \delta$, $N(n)$ satisfies WTA in rounds $[t(n), ..., t(n) + n^c]$.

[6] Throughout, *with high probability* (w.h.p.) refers to events occuring with probability $\geq 1 - 1/n^c$ for constant c. Formally, a family of events $\mathcal{E} = \{E(n)\}$ for all integers $n \geq 1$ occurs w.h.p. if there exists $c > 0$ such that for all n, $\Pr[E(n)] \geq 1 - 1/n^c]$.

In this way, as long as ≥ 2 outputs fire in round t, both inhibitors fire w.h.p. and the high level of inhibition causes outputs to 'drop out of contention' for the winning position with probability $1/2$. After $O(\log n)$ rounds, nearly all the outputs stop firing and with constant probability there is a round in which exactly 1 output fires. Once this round occurs, z_c ceases firing w.h.p. and just z_s fires. This decreased level of inhibition allows the winner to keep firing, as it is offset by the winner's excitatory self-loop. However, it prevents any other output, whose excitatory self-loop is inactive, from firing w.h.p. See Fig. 2 in Appendix B.1 for illustration of the network with its edge weights. We analyze the network in depth in B.1, showing convergence given any input X and initial output configuration Y^0, and yielding:

▶ **Theorem 3.** *There exists a basic WTA network* N *with* $\alpha = 2$ *inhibitors and* $\mathcal{ET}(\mathrm{N}) = O(\log n)$ *and* $\mathcal{HT}(\mathrm{N}) = O(\log^2 n)$.

In Appendix B.1, we show that the network is optimal up to a $\log \log n$ factor and in Appendix B.2 we show that it represents a critical point in the inhibitor-time tradeoff: any network with just one inhibitor requires $\Omega(n^c)$ rounds to solve WTA. Essentially, it is not possible for a single inhibitor to implement the two opposing tasks of stability and convergence.

WTA with $O(\log n)$ Inhibitors

Our second network represents another extreme point of the inhibitor-time tradeoff, using $\alpha = O(\log n)$ inhibitors to achieve $O(1)$ expected convergence time.

The idea is to approximate the ideal behavior in which outputs fire with probability $1/k_t$ in round $t+1$ if k_t outputs fired in round t. As in our two inhibitor algorithm, we have a single stability inhibitor z_s that fires w.h.p. whenever at least one output fires and insures that as soon as a single output fires in a round, the network converges to WTA. We then have $\lceil \log n \rceil - 1$ convergence inhibitors $z_1, ..., z_{\alpha-1}$. We set the bias of the z_i to $b(z_i) = 2^i - .5$ and set $\mathrm{w^{out}}_i = 1$ for all i. In this way, z_i fires w.h.p. in round t whenever $\geq 2^i$ outputs fire. We set the inhibitor to output weights to $\mathrm{w^{inh}}_i = \Theta(\lambda)$ for all i. Thus, when $k_t \in [2^i, 2^{i+1})$, w.h.p. inhibitors $z_1, ..., z_i$ all fire (while $z_{i+1}, ..., z_{\alpha-1}$ do not). The total inhibition from the inhibitors is thus $\Theta(i\lambda)$ and hence each of the k_t outputs fire with probability $1/(1 + e^{\Theta(i)}) \approx 1/2^i \approx 1/k_t$ in round $t+1$. In expectation (and with constant probability) there will be exactly one firing output, giving an expected runtime of just $O(1)$ rounds to reach WTA. In Appendix B.3, we give a full analysis, yielding:

▶ **Theorem 4.** *There exists a basic WTA network* N *with* $\alpha = O(\log n)$ *inhibitors,* $\mathcal{ET}(\mathrm{N}) = O(1)$ *and* $\mathcal{HT}(\mathrm{N}) = O(\log n)$.

Vacuously, no network can beat this expected runtime. We also show in Appendix B.3 that no network can do better with high probability: even with an unlimited number of inhibitors, $\Theta(\log n)$ rounds are requires to solve WTA w.h.p. Intuitively, as long as WTA has not yet been reached in round t, there is no single distinguished output. All outputs have identical connections to X, Z so each active output fires with the *same* probability p in round $t+1$.[7]. Hence the probability that a single output becomes distinguished (is the only one to fire) is $k_t \cdot p(1-p)^{k_t - 1}$, which is bounded by a constant for all k_t, p. Thus, converging to the WTA state w.h.p. takes at least $\Omega(\log n)$ rounds.

[7] By the monotonicity property, it is sufficient to consider in round t only the outputs that fire in round $t-2$, all these outputs have the same firing probability p.

4 WTA with $\alpha \geq 2$ Inhibitors

The above results give a rough outline of the tradeoff between the number of inhibitors and the runtime for WTA. We now explore this tradeoff in more depth for general $\alpha \in (2, \log n]$

4.1 Upper Bound Networks

We first show that both our two inhibitor and $\lceil \log n \rceil$ inhibitor networks can be improved significantly with modest increases in the number of inhibitors or runtime used. We can (up to constant factors) match the runtime of the $\lceil \log n \rceil$ inhibitor network with just $O(\log^{1/c} n)$ inhibitors for any c. Additionally, for any $\alpha \geq \log \log n$ we can achieve expected runtime $O\left(\frac{\log \log n \log \log \log n}{\log \alpha}\right)$, nearly matching our main lower bound of Section 4.2.

▶ **Theorem 5.** *For any integer θ, there is a basic WTA network* N *with $\alpha = O(\theta \log^{1/\theta} n)$ inhibitors, $\mathcal{ET}(\mathrm{N}) = O(\theta)$, and $\mathcal{HT}(\mathrm{N}) = O(\theta \log n)$.*

For $\alpha \geq \log \log n$, writing $\alpha = \log \log^x n$ for $x \geq 1$ if we set $\theta = \frac{c_1 \log \log n \log \log \log n}{\log \alpha} = \frac{c_1 \log \log n}{x}$ then the number of inhibitors required is: $\frac{c_1 \log \log n}{x} \cdot e^{x/c_1} \leq \log \log^x n \leq \alpha$ for small enough c_1.

Proof Sketch. To see the high level idea, consider the case of $\theta = 2$. We will $2\sqrt{\log n}$ inhibitors which are divided into two classes: $\sqrt{\log n}$ coarse inhibitors and $\sqrt{\log n}$ fine inhibitors. The edges from the fine inhibitors to outputs have weight -1 and the edges from coarse inhibitors to outputs have weight $-\sqrt{\log n}$. All the edges from the outputs to the inhibitors have weight 1. We set the bias values of the inhibitors such that: (1) the i^{th} coarse inhibitor fires if the number of active outputs is at least $2^i\sqrt{\log n}$ and (2) the i^{th} fine inhibitor fires if the number of active outputs is at least 2^i. Consider any output density 2^d and let $d' = \lfloor d/\sqrt{\log n} \rfloor$. When 2^d outputs fire in round t, this will excite the first d' coarse inhibitors. As a result, the firing probability for the outputs in round $t+1$ will be approximately $2^{-d' \cdot \sqrt{logn}}$ (ignoring negligible effects from the fine inhibitors). In other words, within a single round the density will be reduced from 2^d to $2^{d-d'\sqrt{\log n}}$ which is a new density in the range $1, 2, 4, ..., 2^{\sqrt{\log n}}$. After this initial round, since at most $2^{\sqrt{\log n}}$ outputs fire, the circuit converges in constant rounds in expectation as the $\sqrt{\log n}$ fine inhibitors can induce probabilities roughly equal to $1/k_t$ just as is done in the $O(\log n)$ inhibitor circuit.

Generalization to larger θ is by repeating the above construction: we have θ levels of increasing coarseness: $[1, 2^{\log^{1/\theta} n}], [2^{\log^{1/\theta} n}, 2^{\log^{2/\theta} n}], ..., [2^{\log^{(\theta-1)/\theta} n}, 2^{\log n}]$. The $\log^{1/\theta} n$ inhibitors at each level ensure that if the number of firing outputs is at level i in round t, it is reduced to level $i-1$ in round $t+1$, yielding $O(\theta)$ expected runtime. We give a full analysis in Appendix B.4. ◀

Our second construction uses similar techniques, but uses just one convergence inhibitor per density class, balancing the time required to move through each density class and the number of classes used. It significantly improves on our two inhibitor algorithm, achieving runtime $O(\log^{1/c} n)$ for any constant c with $O(1)$ inhibitors and $O(\log \log n)$ runtime with $O(\log \log n)$ inhibitors.

▶ **Theorem 6.** *For any $\alpha \geq 2$, there is a basic WTA network* N *with α inhibitors, $\mathcal{ET}(\mathrm{N}) = O\left(\alpha \log^{1/(\alpha-1)} n\right)$ and $\mathcal{HT}(\mathrm{N}) = O\left(\alpha \log^{1+1/(\alpha-1)} n\right)$.*

Proof Sketch. Consider $\alpha = 3$. We have 2 convergence inhibitors: a fine inhibitor z_f and a coarse inhibitor z_c. The inhibitor z_c fires whenever the number of active outputs is at least $2^{\sqrt{\log n}}$, and induces outputs to fire with probability $1/2^{\sqrt{\log n}}$ in the next round. In this way, starting with any density of firing inputs $k_t \in [2^{\sqrt{\log n}}, n]$, within $\sqrt{\log n}$ rounds the density will be reduced to $\leq 2^{\sqrt{\log n}}$. The inhibitor z_f fires whenever at least 2 outputs fire, and induces outputs to fire with probability $1/2$ in the next round. So, within $\sqrt{\log n}$ additional rounds, with constant probability just a single output will remain firing. Again, a full network description for general α and proof is given in Appendix B.4. ◀

4.2 Lower Bound: The Tradeoff Between Inhibitors and Time

We now present our main lower bound which matches Theorem 5 up to $\log\log\log n$ factors.

▶ **Theorem 7.** *For any basic WTA network* N *with* α *inhibitors,* $\mathcal{ET}(\mathrm{N}) = \Omega\left(\frac{\log n \log n}{\log \alpha}\right)$ *and* $\mathcal{HT}(\mathrm{N}) = \Omega\left(\frac{\log\log n}{\log \alpha} \cdot \frac{\log n}{\log\log\log n}\right)$.

Lower Bound Overview

We focus on initial output configuration $Y^0 = \vec{0}$ (i.e., no output fires in the sub-round $(0, 2)$) which we call the *reset configuration*. We show that for any network N with α inhibitors there exists at least one input X for which the expected time to reach WTA starting from the reset configuration is $\Omega(\log\log n/\log\alpha)$. If suffices to consider the case where $\alpha = O(\log^{1/c} n)$ for some constant c since for $\alpha = \Theta(\log^{1/c} n)$, the expected runtime is $O(1)$. Throughout this section, we say an event happens with *good probability* if its probability is at least $1 - O(\log^4 n)$.

Our argument contains two main parts. First, we show that the inhibitors fire in a *nearly* deterministic manner and hence we can treat them (up to some slack) as *threshold circuits*. Equipped with this property, we then consider $\Theta(\log n)$ *density classes* each covering a constant multiplicative range of firing outputs. The predictable behavior of the inhibitors is used to show that even after $\Omega(\log n \log n/\log\alpha)$ rounds, the network cannot distinguish between at least two different density classes, which yields our claim as it does not converge to WTA for at least one class.

(1) Inhibitor classification: inhibitors are nearly deterministic for most density classes

To address the first challenge (i.e., showing that inhibitors are predictable), we divide the set of inhibitors Z into three classes and show the predictability property for each class separately. The "stability" class (or "WTA preservers") S contains inhibitors whose *goal* is to maintain the WTA steady state. The "convergence" class (or "progress inhibitors") C contains the inhibitors that are responsible for driving fast convergence to a WTA state. Finally, the third class R contains the remaining inhibitors whose contribution to both stability and convergence is negligible.

Formally, for any inhibitor $z_i \in \mathrm{Z}$ and $j \in [1, n]$ let $pot_j(z) = j \cdot \mathrm{w^{out}}_i - b(z_i)$ be the potential of z_i when exactly j outputs fire (I.e., if in sub-round $(t, 2)$ the number of firing outputs is j, then the potential of z_i in sub-round $(t, 3)$ is $pot_j(z)$ and it fires in sub-round $(t, 3)$ with probability $1/(1 + e^{-pot_j(z)})$). The set S contains all inhibitors that fire in steady state (i.e., when exactly one output is firing) with reasonably high probability. Fixing some constant $c \geq 1$, $S = \{z_i \in \mathrm{Z} \mid 1/(1 + e^{-pot_1(z_i)}) \geq 1/\log^{3c} n\}$. The set C is comprised of

all inhibitors $z_i \notin S$ whose firing probability is least $1/\log^c n$ when all n outputs fire in the previous sub-round: $C = \{z_i \in \mathrm{Z} \mid z_i \notin S \text{ and } 1/(1+e^{-pot_n(z_i)}) \geq 1/\log^c n\}$[8]. Finally, R contains all remaining inhibitors not in S or C.

We show that the firing states of the inhibitors can *in certain cases* be predicated with good probability. The argument for each of the three classes S, C and R is different and is presented in Appendix B.5.1. Since the inhibitors in S fire with good probability when just one output fires, we can show that they fire w.h.p. when at least two outputs fire:

▶ **Lemma 8** (S is predictable). *Let $(t,2)$ be a sub-round in which at least two outputs fire, then sub-round $(t,3)$, all inhibitors of S fire with probability at least $1-1/n$.*

Since the firing probability of the R inhibitors is small compared to the $O(\log\log n/\log\alpha)$ execution length that we care about, we have:

▶ **Lemma 9** (R is predictable). *Given any input* X *and any initial configuration, with probability at least $1-1/\log^{c-3} n$, none of the inhibitors in R fire in $O(\log^2 n)$ rounds of execution of* N.

Perhaps the most surprising claim concerns the predictability of the convergence inhibitors:

▶ **Lemma 10** (C is almost predictable). *For every $z \in C$, there exists an integer $k(z) \in [1,n]$, such that for $c \geq 4$:*

(1) Low Density: *When there are at most $k(z)/2$ firing outputs in sub-round $(t,2)$, the probability that z fires in sub-round $(t,3)$ is at most $1/\log^c n$ (i.e., with good probability, z does not fire);*

(2) High Density: *When there are at least $2k(z)$ firing outputs in sub-round $(t,2)$, the probability that z fires in sub-round $(t,3)$ is at least $1-1/\log^c n$ (i.e., with good probability, z fires).*

Overall, except for the case where the number of firing outputs in sub-round $(t,2)$ is in the density class $K(z) = [k(z)/2, k(z)]$, z behaves in sub-round $(t,3)$ in an almost deterministic manner. Roughly speaking, this is shown by exploiting the *gap* in the firing probabilities of these inhibitors between the steady state rounds (when they fire with probability $\leq 1/\log^{3c} n$) and the rounds in which there are sufficiently many firing outputs (where they fire with probability $\geq 1/\log^c n$). The proof of Lemma 10 shows that this gap implies that the sigmoid function which converts the number of firing inputs to z's firing probability must be steep enough such that z has predictable behavior outside a small range around $k(z)$.

(2) Network prediction for nearly deterministic inhibitors

Using the predictable nature of the inhibitors, we now show that there is at least one *density class* of competing inputs for which we can predict (with good probability) the behavior of N for $\Omega(\log\log n/\log\alpha)$ rounds, at the end of which the WTA state has not been reached. We consider a set of $\ell = \lfloor \log n \rfloor$ inputs $\mathcal{X} = \{\mathrm{X}_1, ..., \mathrm{X}_\ell\}$ where X_i contains exactly 2^i firing inputs (i.e. $\|\mathrm{X}_i\|_1 = 2^i$). Thus, $\mathcal{X}$ contains a representative input from each density class of input vectors whose number of firing inputs is within a factor two of each other.

For any $\mathrm{X} \in \mathcal{X}$ let $\widehat{R}_t(\mathrm{X}) \in \{1,\ldots,n\}$ be the random variable indicting the number of firing outputs in sub-round $(t,2)$ starting from the initial configuration $Y_0 = \vec{0}$. Let $\widehat{F}_t(\mathrm{X}) \in \{0,1\}^\alpha$ be the random variable indicating the firing status of the inhibitors in

[8] The difference between $1/\log^{3c} n$ when defining the threshold for the inhibitors in S and $1/\log^c n$ when defining the threshold for the inhibitors C, is crucial in the analysis.

sub-round $(t, 3)$. For each $\mathrm{X} \in \mathcal{X}$ we will attempt to maintain a *predicted* range $R_t(\mathrm{X})$ of the number of firing outputs in sub-round $(t, 2)$ along with a *predicted* inhibitor configuration in sub-round $(t, 3)$, $F_t(\mathrm{X})$. We will let $\mathcal{X}_t \subseteq \mathcal{X}$ denote the subset of inputs whose behavior we can predict well in (all sub-rounds of) round t – specifically, for which we know $\widehat{R}_t(\mathrm{X}) \in R_t(\mathrm{X})$ and $\widehat{F}_t(\mathrm{X}) = F_t(\mathrm{X})$ with good probability (at least $1 - 1/\log n$).

For any inhibitor $z \in C$, we call the range $K(z) = [k(z)/2, 2k(z)]$– the *critical range* of z (see Lemma 10 for the definition of $k(z)$). If the number of firing outputs enters this range, we will not be able to predict the behavior of z in the next sub-round with good probability. On the other hand, as long as the number of firing outputs in sub-round $(t, 2)$ is not in the critical range of any $z \in C$, then the firing behavior of the inhibitors in sub-round $(t, 3)$ can be predicted with good probability.

We will progress through rounds, predicting the behavior of N in round t for each input in $\mathcal{X}_{t-1}$ based off the predictions in round $t - 1$. We will ensure that in any round, not too may inputs have predicted ranges overlapping critical regions by ensuring that these predicted ranges remain separated by constant factors and hence, at most $|C|$ of them can overlap $K(z)$ for some $z \in C$.

Predicting the number of firing outputs given inhibitor states

We now describe how to predict the range $R_t(X)$ given the prediction $F_{t-1}(X)$. Our main goal is to preserve the separation between the predicted ranges $R_t(\mathrm{X})$ for sufficiently many inputs $\mathrm{X} \in \mathcal{X}_{t-1}$.

To maintain the separation, we consider only the largest subset $\mathcal{X}_t^{same} \subseteq \mathcal{X}_{t-1}$ of inputs whose predicted firing configuration for the inhibitors in the previous sub-round $(t - 1, 3)$ is exactly the *same* (i.e., inputs X with the same $F_{t-1}(\mathrm{X})$ vector). By doing this, we guarantee that the firing probabilities of all the outputs in sub-round $(t, 2)$ is the same. Letting this probability be p, the expected number of firing outputs in sub-round $(t, 2)$ is in the range $p \cdot R_{t-1}(\mathrm{X})$ for each $\mathrm{X} \in \mathcal{X}_t^{same}$ and the separation between these ranges is preserved in expectation. To show that the ranges are also separated with good probability, we omit from $\mathcal{X}_t^{same}$ at most $\Theta(\log \log n)$ inputs with ranges $R_t(\mathrm{X})$ containing values $\leq \log^c n$ for some constant c. They remaining inputs thus have output ranges concentrated around their expectation. The key point to observe is that because the inhibitors behave almost as threshold circuits, the number of different firing configurations in sub-round $(t - 1, 3)$ is at most α (i.e., there are at most α different $F_{t-1}(\mathrm{X})$ vectors for $\mathrm{X} \in \mathcal{X}_{t-1}$) and hence the cardinality of the set $\mathcal{X}_t^{same}$ for which we predict the range of firing outputs in sub-round $(t, 2)$ is at least $|\mathcal{X}_{t-1}|/\alpha$.

Predicting the inhibitor states given the number of firing outputs

We next describe how to predict the inhibitor firings $F_t(\mathrm{X})$ given the prediction $R_t(\mathrm{X})$. Since the convergence inhibitors are predictable when the number of firing outputs is not in any critical range $K(z)$, we first omit from $\mathcal{X}_t^{same}$ all inputs X whose predicted range $R_t(\mathrm{X})$ intersects the critical range of some $z \in C$ (i.e. $R_t(\mathrm{X}) \cap K(z) \neq \emptyset$ for some z). We call the resulting set $\mathcal{X}_t$. Since the ranges of $\mathcal{X}_t^{same}$ are separated by some constant, we do not discard more than $|C| = O(\alpha)$ inputs.

Overall, we predict the circuit behavior in sub-rounds $(t, 2), (t, 3)$ with good probability for all inputs $\mathrm{X} \in \mathcal{X}_t$ where $|\mathcal{X}_t| \geq |\mathcal{X}_{t-1}|/\alpha - \alpha$. Since $\alpha = O(\log^{1/c} n)$, we get that after t rounds, there are $|\mathcal{X}_t| = \Omega(\log n/\alpha^t)$ inputs for which the network behaves *exactly* the same in each of the t rounds with good probability. This argument proceeds as long as

$\log n/\alpha^t \geq 2$, leading to the lower bound of expected time $\Omega(\log\log n/\log\alpha)$ since we can show if two inputs are not distinguished, at least one will not have reached WTA. In Appendix B.5.2, we describe the prediction process in detail and complete the proof of Theorem 7.

High Probability Lower Bound

Finally, we show that our lower bound for expected runtime extends to a lower bound on the high probability runtime. Our lower bound implies that "repeating" the execution of a network that converges with constant probability $\Theta(\log n)$ times to achieve a high probability guarantee is essentially the best one can do (up to a $\log\log\log n$ factor).

▶ **Lemma 11.** *For any basic WTA network* N *with* α *inhibitors* $\mathcal{HT}(\mathrm{N}) = \Omega(\frac{\log n\cdot\log\log n}{\log\alpha\log\log\log n})$.

Proof Sketch. Let $DC = \Theta\left(\frac{\log\log n}{\log\alpha}\right)$ and $DH = DC\cdot\left(\frac{\log n}{\log\log\log n}\right)$. Fix a network N with α inhibitors and let X be the input for which, by Theorem 7, N requires at least DC rounds in expectation starting from initial configuration $\mathcal{C}_0$ with input X and $\mathrm{Y}^0 = \vec{0}$. In the following proof, we will actually exploit the fact that the lower bound in Theorem 7 applies to the time it takes to reach a WTA state with *constant probability* (a stronger time measure than expected time).

We work with the *execution tree* T which includes all possible DH round executions of N starting from $\mathcal{C}_0$. The tree T has depth DH where each layer corresponds to the configuration of the network in each round t. Each node u at level t is labeled by an $(n+\alpha)$-length binary vector $Q(u)$ describing the firing states of the outputs and inhibitors in round t, i.e., the firing states of the outputs in sub-round $(t,2)$ and the firing states of the inhibitors in sub-round $(t,3)$. Node u has $2^{n+\alpha}$ children, with the edge to each child labeled with the transition probability between the configuration in u to the child configuration. The root node r is labeled with $\mathcal{C}_0$. The mass of node u is given by the product of edge weights on its path to r. It is the probability of reaching u's configuration through that execution path. We call a node u a *reset* node (resp., *WTA* node), if in the configuration $Q(u)$ no output fires (resp., exactly one output with active input fires).

To lower bound $\mathcal{HT}(\mathrm{N})$ we will show that the probability to reach a non-WTA leaf node when starting from the root r is at least $1/n^2$, and thus the probability to reach a WTA leaf node is at most $1-1/n^2 < 1-1/n^c$, contradicting a w.h.p. runtime of $\leq DH$ rounds.

Our strategy is based on traversing the tree in an asynchronous manner from the root to (sufficiently many) non-WTA leaf nodes with sufficiently high total probability mass. For a given node u in layer t, we may move to a subset of its *non-WTA* children nodes in layer $t+1$. We call this move a *small jump*. Alternatively, we may make a *large jump*, moving DC steps from u and proceeding the traversal from a subset of *non-WTA* leaf nodes of $T_{DC}(u)$ (the height DC subtree rooted at u). With each jump starting at u, we loose some probability mass – the idea is to show that we do not loose it too quickly.

In more detail, in each step of our traversal, we maintain a collection of non-WTA nodes. When arriving a node u in the traversal, we consider its configuration $Q(u)$ and look at the probability that the next round is a *reset* round (with 0 firing outputs) given $Q(u)$. We show that if the probability of having at most 1 firing outputs in the next round is $\geq 1/\log\log n$, the probability of having a reset (no firing outputs) is large – i.e., $\geq 1/(\log\log n)^3$.

In this case we continue traversal only from the children of u that are reset nodes. For each of these children v, let $T_{DC}(v)$ be the execution tree of depth DC rooted at v. By the lower bound in Theorem 7, the probability to reach a non-WTA leaf node in $T_{DC}(v)$ starting from $Q(v)$ is at least a constant. So from each reset-node v, we make a large jump to the leaves of

$T_{DC}(v)$. Overall, we maintain a $\Theta(1/(\log\log n)^3)$ fraction of the probability mass of u in making this large jump. Since such a jump can occur at most $DH/DC = \log n/\log\log\log n$ times, we maintain at least a $1/(\log\log n)^{3DH/DC} \geq 1/n^2$ fraction of the probability mass throughout the traversal.

On the other hand, when arriving a node u for which the probability of having at most 1 firing output in the next round is less than $1/\log\log n$, we make a small jump to the children of u in which the number of firing outputs is at least 2 (and hence which are non-WTA nodes). This jump maintains $1 - 1/\log\log n$ of the probability mass and since such a jump can happen at most DH times, we again maintain $(1-1/\log\log n)^{DH} \geq 1/n^2$ of the original probability. Overall, through making both large and small jumps, at the end of the traversal, we reach a set of non-WTA nodes containing at least a $1/n^2$ fraction of the probability mass in the DH level execution tree. This gives us our high probability time lower bound. See Appendix B.6 for a complete analysis and Fig. 3 for an illustration of the execution tree. ◀

In Appendix C, we extend our lower bounds (for both expected and high probability time) to the case where the α auxiliary neurons can be both excitatory and inhibitory neurons. The more general bound holds under the restriction that outputs with no active input are not allowed to fire during the execution. Only competing outputs (that have a positive signal from their inputs) ever fire.

5 Discussion

We hope that this paper is a starting point for further investigation into stochastic spiking networks from an algorithmic perspective, which investigates fundamental tradeoffs between biological resources and identifies basic building blocks and principles for algorithm design in neural settings.

We focus on a restricted class of three layer networks, in which auxiliary neurons are not interconnected. This models the generally restricted connectivity structure that inhibitory neurons appear to have in biological networks and lets us give both very strong upper bounds and matching lower bounds. Still, it would be interesting to understand the effect of connections between auxiliary neurons. We have preliminary work showing that some speedups are possible in these more general networks, however obtaining any non-trivial lower bounds would be very interesting.

Studying other important primitives aside from the binary version of WTA that we focus on would also be interesting. We again have preliminary work on *non-binary WTA* in which the network must choose the input with the highest, or near highest firing rate as the winner. There are many other problems to consider.

Our model attempts to be biologically plausible enough to capture high level behavior, yet not be overly complex. However, many modeling assumptions are possible, and we hope that future work explores if changes to the model can lead to significant differences in computational power or algorithmic techniques. As an example, for simplicity we considered a synchronous model, however, asynchrony seems to be an important part of neural computation which would be valuable to study.

Finally, we note that significant theoretical work attempts to understand how neural networks can *learn* through the modification of synapse weights as their endpoints fire more or less frequently [46, 36]. The most common model for how synapse weights evolve is the *hebbian learning* rule, which is itself the focus of a vast literature. Merging the view of neural networks as executing algorithms given predetermined network parameters with understanding of learning would be very interesting. Can a WTA network 'evolve' naturally

via simple learning rules? How do fixed network motifs such as WTA circuits interact with more flexible 'learning' networks?

Acknowledgments. We are grateful to Mohsen Ghaffari for noting the general upper bound network construction and for many helpful discussions on the lower bound proof. We would also like to thank Nir Shavit, Rati Gelashvili, and Sergio Rajsbaum for insightful discussions.

References

1 David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169, 1985.

2 Maruan Al-Shedivat, Rawan Naous, Emre Neftci, Gert Cauwenberghs, and Khaled N Salama. Inherently stochastic spiking neurons for probabilistic neural computation. In *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 356–359. IEEE, 2015.

3 Christina Allen and Charles F Stevens. An evaluation of causes for unreliability of synaptic transmission. *Proceedings of the National Academy of Sciences*, 91(22):10380–10383, 1994.

4 Sander M Bohte, Joost N Kok, and Han La Poutre. Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, 48(1):17–37, 2002.

5 Romain Brette, Michelle Rudolph, Ted Carnevale, Michael Hines, David Beeman, James M Bower, Markus Diesmann, Abigail Morrison, Philip H Goodman, Frederick C Harris Jr, et al. Simulation of networks of spiking neurons: a review of tools and strategies. *Journal of computational neuroscience*, 23(3):349–398, 2007.

6 Lars Buesing, Johannes Bill, Bernhard Nessler, and Wolfgang Maass. Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons. *PLoS Comput Biol*, 7(11):e1002211, 2011.

7 Robert Coultrip, Richard Granger, and Gary Lynch. A cortical model of winner-take-all competition via lateral inhibition. *Neural networks*, 5(1):47–54, 1992.

8 Shlomi Dolev. *Self-stabilization*. MIT press, 2000.

9 Shlomi Dolev, Amos Israeli, and Shlomo Moran. Uniform dynamic self-stabilizing leader election. *IEEE Transactions on Parallel and Distributed Systems*, 8(4):424–440, 1997.

10 A Aldo Faisal, Luc PJ Selen, and Daniel M Wolpert. Noise in the nervous system. *Nature reviews neuroscience*, 9(4):292–303, 2008.

11 Michael Fischer and Hong Jiang. Self-stabilizing leader election in networks of finite-state anonymous agents. In *International Conference On Principles Of Distributed Systems*, pages 395–409. Springer, 2006.

12 Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

13 Sonia M Gómez-Urquijo, Concepción Reblet, José L Bueno-López, and Iñaki Gutiérrez-Ibarluzea. Gabaergic neurons in the rabbit visual cortex: percentage, layer distribution and cortical projections. *Brain research*, 862(1):171–179, 2000.

14 Ankur Gupta and Lyle N Long. Hebbian learning with winner take all for spiking neural networks. In *2009 International Joint Conference on Neural Networks*, pages 1054–1060. IEEE, 2009.

15 Stefan Habenschuss, Zeno Jonke, and Wolfgang Maass. Stochastic computations in cortical microcircuit models. *PLoS Comput Biol*, 9(11):e1003311, 2013.

16 John J Hopfield, David W Tank, et al. Computing with neural circuits- a model. *Science*, 233(4764):625–633, 1986.

17 Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3):194–203, 2001.

18 Eugene M Izhikevich. Which model to use for cortical spiking neurons? *IEEE transactions on neural networks*, 15(5):1063–1070, 2004.

19 Zeno Jonke, Stefan Habenschuss, and Wolfgang Maass. Solving constraint satisfaction problems with networks of spiking neurons. *Frontiers in neuroscience*, 10, 2016.

20 Samuel Kaski and Teuvo Kohonen. Winner-take-all networks for physiological models of competitive learning. *Neural Networks*, 7(6-7):973–984, 1994.

21 Christof Koch and Shimon Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence*, pages 115–141. Springer, 1987.

22 John Lazzaro, Sylvie Ryckebusch, Misha Anne Mahowald, and Caver A Mead. Winner-take-all networks of o (n) complexity. Technical report, DTIC Document, 1988.

23 Dale K Lee, Laurent Itti, Christof Koch, and Jochen Braun. Attention activates winner-take-all competition among visual filters. *Nature neuroscience*, 2(4):375–381, 1999.

24 Nancy Lynch. A hundred impossibility proofs for distributed computing. In *Proceedings of the eighth annual ACM Symposium on Principles of distributed computing*, pages 1–28. ACM, 1989.

25 Nancy A Lynch. *Distributed algorithms*. Morgan Kaufmann, 1996.

26 Wolfgang Maass. On the computational power of noisy spiking neurons. *Advances in neural information processing systems*, pages 211–217, 1996.

27 Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

28 Wolfgang Maass. Neural computation with winner-take-all as the only nonlinear operation. In *NIPS*, pages 293–299. Citeseer, 1999.

29 Wolfgang Maass. On the computational power of winner-take-all. *Neural computation*, 12(11):2519–2535, 2000.

30 Wolfgang Maass. Noise as a resource for computation and learning in networks of spiking neurons. *Proceedings of the IEEE*, 102(5):860–880, 2014.

31 Marvin Minsky and Seymour Papert. Perceptrons. 1969.

32 Steven J Nowlan. Maximum likelihood competitive learning. In *NIPS*, pages 574–582, 1989.

33 Matthias Oster, Rodney Douglas, and Shih-Chii Liu. Computation with spikes in a winner-take-all network. *Neural computation*, 21(9):2437–2465, 2009.

34 Matthias Oster and Shih-Chii Liu. Spiking inputs to a winner-take-all network. *Advances in Neural Information Processing Systems*, 18:1051, 2006.

35 Christos H Papadimitriou and Santosh S Vempala. Unsupervised learning through prediction in a model of cortex. *arXiv preprint arXiv:1412.7955*, 2014.

36 Christos Papadimitrou, Samantha Petti, and Santosh Vempala. Cortical computation via iterative constructions. *arXiv preprint arXiv:1602.08357*, 2016.

37 Josep L Rossello, Vincent Canals, Antoni Morro, and Antoni Oliver. Hardware implementation of stochastic spiking neural networks. *International journal of neural systems*, 22(04):1250014, 2012.

38 Lisa Roux and György Buzsáki. Tasks for inhibitory interneurons in intact brain circuits. *Neuropharmacology*, 88:10–23, 2015.

39 Bernardo Rudy, Gordon Fishell, SooHyun Lee, and Jens Hjerling-Leffler. Three groups of interneurons account for nearly 100% of neocortical gabaergic neurons. *Developmental neurobiology*, 71(1):45–61, 2011.

40 BL Sabatini and WG Regehr. Timing of synaptic transmission. *Annual Review of Physiology*, 61(1):521–542, 1999.

41 H Sebastian Seung. Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron*, 40(6):1063–1073, 2003.

42 Michael N Shadlen and William T Newsome. Noise, neural codes and cortical organization. *Current opinion in neurobiology*, 4(4):569–579, 1994.

43 Simon J Thorpe. Spike arrival times: A highly efficient coding scheme for neural networks. *Parallel processing in neural systems*, pages 91–94, 1990.

44 Leslie G Valiant. *Circuits of the Mind*. Oxford University Press on Demand, 2000.

45 Leslie G Valiant. A neuroidal architecture for cognitive computation. *Journal of the ACM (JACM)*, 47(5):854–882, 2000.

46 Leslie G Valiant. Memorization and association on a realistic neural model. *Neural computation*, 17(3):527–555, 2005.

47 Wei Wang and Jean-Jacques E Slotine. K-winners-take-all computation with neural oscillators. *arXiv preprint q-bio/0401001*, 2003.

48 Alan L Yuille and Norberto M Grzywacz. A winner-take-all mechanism based on presynaptic inhibition feedback. *Neural Computation*, 1(3):334–347, 1989.

A Additional Discussion

A.1 Related Work

Spiking Neural Networks

A vast literature studies computation in stochastic spiking neural networks. Work includes detailed models aimed at matching biological observations [12, 18], large scale simulation in hardware and software [5, 37], attempts to understand general properties of computation in these networks [6], the design of specific algorithms [4, 41], and theoretical investigation of computational power [26, 15]. For instance, it has been shown that deterministic spiking networks can simulate Turing machines and that stochastic spiking networks can implement MCMC sampling [6]. As is popular in the biologically-inspired algorithms literature, spiking networks have been used as heuristic 'stochastic search' solvers for NP-hard constraint satisfaction problems, such as Sudoku and TSP [19].

Our model can be seen as a discrete version of the continuous model discussed in by Maass in [30] or as a noisy version of the deterministic model in [27]. In addition to being stochastic, in comparison to the model of [27], our response latency Δ is constant for all connections in the network. Additionally, we have just a single round memory – each neuron's membrane potential is affected just by spikes of neighboring neurons in the same or immediately preceding round of computation. We note that if connections are allowed between auxiliary neurons, a longer memory can be easily be implemented within our general model.

Self-Stabilization in Distributed Computing

The notion of self-stabilization goes back to Dijkstra in 1973. A self-stabilizing system can automatically recover following the occurrence of transient faults. The goal in this area is to design systems that converge to a desired behavior from any arbitrary starting point [8, 25]. Among the tremendously broad work, perhaps the most relevant to this work is self-stabilizing algorithms for leader election [9, 11].

In a stochastic neural network, self-stabilization is a necessity. Both changes to the given input as well as random deviations of the system from a converged state require the network to re-converge. Hence, we insure that all our networks converge to WTA from any initial network configuration and are self-stabilizing. This property does not hold in many previously studied WTA implementations for spiking networks [33].

Valiant's Neuroidal Model

Valiant considers a model of neural computation in which abstract neurons (which he calls *neuroids*) are connected via a random network of synapses [44]. He discusses how these neurons can learn representations of real world objects whose perception stimulates the network in certain ways. As in our model, neurons fire in response to a membrane potential given by a weighted sum of firing neighbors. Differently, synapse weights evolve in response to increased firing of their end points, which allows *learning* to occur within the network. This learning ability is the primary focus of Valiant's work and of follow up work on the model. For example, recently, [35] extended understanding of how reasonably complex learning and pattern matching tasks can be performed in this model.

Our work deviates is somewhat more 'algorithmic' than the work of Valiant, focusing how basic takes can be computed using a set of neurons with a fixed set of synapses and bias values. We do not consider how, for example, our WTA networks could form within a larger neural circuit through learning of appropriate synapse weights. Following previous work [28] we think of WTA networks as fundamental primitives of neural circuits on top of which high level algorithms, such as learning algorithms, can be built.

A.2 Biological Motivation for Network Dynamics

The timing of neural spikes is determined by two biological parameters, namely, the *refractory period* β and the *response latency*, Δ. The refectory period is the time during which stimulus given to the neuron would not cause a second action potential. The response latency is the delay between the time the action potential reaches the presynaptic terminal of the input neurons and the time the postsynaptic output neuron sends out an action potential (assuming it does). In our setting we consider the case where $\Delta < \beta$ since for connected neurons in close proximity to each other, and inhibitory neurons with primarily local connections, the response delay is a few hundred of micro-seconds whereas the refractory time is several milliseconds [40]. WTA networks are basic, local neural primitives that are not believed to involve long range connections, justifying our assumption.

Every round corresponds to an interval between two pulses of the inputs (hence a round lasts β milliseconds). At the beginning of every round, the input layer spikes (at sub-round $(t, 1)$ in the notation of our discrete model). The spikes generated by the inputs invoke an alternating dynamic between the three layers in the circuit. Specifically, with a delay of δ milliseconds after the input's spike, the outputs spike with probability proportional to their total synaptic strengths (in sub-round $(t, 2)$). As shown in equation (1), this potential incorporates any spikes which occurred within a β millisecond preceding window – the input spikes in sub-round $(t, 1)$ (Δ milliseconds before), the inhibitor spikes in sub-round $(t-1, 3)$ ($\beta - \Delta$ milliseconds before), and the neuron's own self-excitatory output spike in sub-round $(t-1, 2)$, β milliseconds before. Δ milliseconds after the outputs spike, the inhibitors spike in sub-round $(t, 3)$, again incorporating spikes that occurred with a β millisecond window, which due to their limited connectivity structure, just includes the spikes of Y in $(t, 2)$.

B Missing Proofs and Auxiliary Claims

Throughout, we make use of the following Corollary of the Chernoff bound.

▶ **Theorem 12** (Simple Corollary of Chernoff Bound). *Suppose $X_1, X_2, \ldots, X_\ell \in [0, 1]$ are independent random variables. Let $X = \sum_{i=1}^{\ell} X_i$ and $\mu = \mathbb{E}[X]$. If $\mu \geq 5 \log n$, then w.h.p. $X \in \mu \pm \sqrt{5\mu \log n}$, and if $\mu < 5 \log n$, then w.h.p. $X \leq \mu + 5 \log n$.*

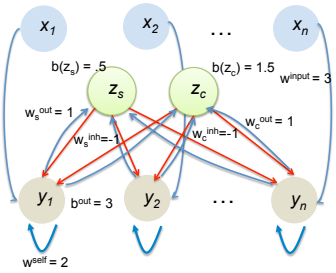


Figure 2 Two Inhibitor WTA Network.

B.1 WTA with Two Inhibitors

Proof of Theorem 3 (Two Inhibitor Upper Bound)

Formally the parameters of the network are set as follows: assume w.l.o.g. that $\lambda = 1/(c_1 \log n)$ for large constant c_1. For both inhibitors, set the excitatory output to inhibitor weights to $\mathrm{w}^{\mathrm{out}}{}_s = \mathrm{w}^{\mathrm{out}}{}_\ell = 1$ and $b(z_s) = .5$, $b(z_c) = 1.5$. Thus, by equation (2) z_s fires w.h.p. in sub-round $(t, 3)$ whenever at least one output fires in sub-round $(t, 2)$, and z_c fires w.h.p. whenever at least two outputs fire.

Set the inhibitor to output weights to $\mathrm{w}^{\mathrm{inh}}{}_s = \mathrm{w}^{\mathrm{inh}}{}_\ell = -1$, the excitatory input to output connection weight to $\mathrm{w}^{\mathrm{input}} = 3$, and the excitatory output to output self-loop to $\mathrm{w}^{\mathrm{self}} = 2$. Finally, set the output bias to $\mathrm{b}^{\mathrm{out}} = 3$.

The above parameters insure that only outputs corresponding to firing inputs ever fire w.h.p. Additionally, if we have not yet reached WTA and both z_s and z_c fire in sub-round $(t, 3)$, any output that fired in sub-round $(t, 2)$ will fire with probability $1/2$ in sub-round $(t + 1, 2)$. If we have reached WTA and just z_s fires, any output (the winner) that fired in round t will fire in round $t + 1$ w.h.p. In either case, any output that did not fire in round t will not fire w.h.p. in round $t + 1$.

We now give a formal proof of the theorem. First note that if the input $\mathrm{X} = \vec{0}$ then in every round, each output has potential $pot(y_j, t) \leq \mathrm{w}^{\mathrm{self}} - \mathrm{b}^{\mathrm{out}} = -1$ and so, recalling that $\lambda = 1/(c_1 \log n)$, fires with probability at most $\frac{1}{1+e^{c_1 \log n}} \leq 1/n^c$ for some large constant c in any round. So w.h.p. no outputs fire in each round, which is the valid output given $X = \vec{0}$ and so N trivially converges to WTA. So for the remainder of the section we focus on the case in which X has at least one firing input. We show that N satisfies the following conditions, which imply Theorem 3:

▶ **Claim 13** (Stability). *If* N *satisfies WTA in round* t *with* $y_j^t = 1$*, then* N *satisfies WTA in round* $t + 1$ *with* $y_j^{t+1} = 1$ *w.h.p.*

▶ **Claim 14** (Convergence). *Letting* $t = c_2 \log n$ *for constant* c_2*, for any input* X *with* $\|X\|_1 \geq 1$ *and any starting configuration* C^0*,* N *satisfies WTA in round* $C^{t'}$ *for some* $t' < t$*, with constant probability.*

Since Claim 14 holds for any starting configuration, we can simply apply it $\Theta(\log n)$ times to show that w.h.p. within $\Theta(\log^2 n)$ rounds, there will be a round in which WTA is satisfied, and hence N will converge to WTA by Claim 13. Additionally, it gives $\mathcal{ET}(\mathrm{N}) = O(\log n)$ as letting c_1 be the constant probability of reaching WTA in $O(\log n)$ rounds, we have:

$$\mathcal{ET}(\mathrm{N}) = O\left(\sum_{i=0}^{\infty}(1 - c_1)^i \cdot c_1 \log n\right) = O(\log n).$$

This gives us Theorem 3.

Proof of Claim 13. N satisfies WTA in round t with output y_j firing, so we have

$$pot(z_s, t) = 1 \cdot \mathrm{w^{out}}_s - b(z_s) = .5 \text{ and } pot(z_c, t) = 1 \cdot \mathrm{w^{out}}_s - b(z_c) = -.5.$$

Thus, recalling that $\lambda = 1/(c_1 \log n)$, in round t z_s fires with probability $\frac{1}{1+e^{-.5c_1 \log n}} \geq 1 - 1/n^c$ for large c and z_c fires with probability $\frac{1}{1+e^{.5c_1 \log n}} \leq 1/n^c$ for large c. So w.h.p. just z_s fires in round t. This gives that w.h.p.

$$pot(y_j, t+1) = (1 \cdot \mathrm{w^{inh}}_s) + (0 \cdot \mathrm{w^{inh}}_\ell) + (1 \cdot \mathrm{w^{self}}) + \mathrm{w^{input}} - \mathrm{b^{out}} = -1 + 2 + 3 - 3 = 1.$$

So y_j fires with probability $\frac{1}{1+e^{c_1 \log n}} \geq 1 - 1/n^c$ in round $t+1$. In contrast, for any $j' \neq j$, $y_{j'}$ does not fire in round t so we have w.h.p.

$$pot(y_{j'}, t+1) \leq (1 \cdot \mathrm{w^{inh}}_s) + (0 \cdot \mathrm{w^{inh}}_\ell) + (0 \cdot \mathrm{w^{self}}) + \mathrm{w^{input}} - \mathrm{b^{out}} = -1 + 3 - 3 = -1.$$

Therefore y'_j fires with probability $\leq 1/n^c$ in round $t+1$ so WTA is satisfied with output y_j firing in round $t+1$ w.h.p. ◀

Proof of Claim 14. Recall that we only consider $\|X\|_1 \geq 1$ as convergence to WTA is trivial when $X = \vec{0}$. We analyze three simple cases depending the initial configuration C^0:

Case 0: No output y_j with $x_j = 1$ fires in $\mathbf{Y^0}$

We first consider the subcase that no output (regardless of the value of x_j) fires in Y^0. In this case, $pot(z_s, 0) = -b(z_s) = -.5$ and $pot(z_c, 0) = -b(z_c) = -1$ so neither inhibitor fires w.h.p. in round 0. So w.h.p. all outputs with firing inputs have $pot(y_j, 1) \geq \mathrm{w^{input}} - \mathrm{b^{out}} = 0$ and so fire with probability $\geq 1/2$ in round 1. Since, $\mathrm{X} \neq \vec{0}$, with constant probability at least one of these outputs fires in round 1, in which case we appeal to Cases 1 and 2 below (where we re-label C^2 as the initial configuration C^0.).

Next consider the case when at least one output fires in Y^0, but all firing outputs correspond to non-firing inputs. In this case, we have $pot(z_s, 0) \geq 1 \cdot \mathrm{w^{out}} - b(z_s) \geq .5$ and so z_s fires w.h.p. in round 0. As noted, in any round, any output y_j with $x_j = 0$ has $pot(y_j, t) \leq \mathrm{w^{self}} - \mathrm{b^{out}} = -1$ and so does not fire w.h.p. Additionally, since every output with $x_j = 1$ has $y_j^0 = 1$, these outputs have $pot(y_j, 1) = \mathrm{w^{inh}}_s + \mathrm{w^{input}} - \mathrm{b^{out}} = -1 + 3 - 3 = -1$ and so do not fire w.h.p. in round 1. So w.h.p. in round 1 no outputs fire and we are in the first case above.

Case 1: Exactly one output y_j with $x_j = 1$ fires in $\mathbf{Y^0}$

By Claim 13 and the fact that outputs with $x_j = 0$ do fire w.h.p. in any round, N satisfies WTA in round 1 and so immediately converges to WTA.

Case 2: More than one output y_j with $x_j = 1$ fires in $\mathbf{Y^0}$

Let k_t be the number of *active* outputs in round t – that is outputs corresponding to firing inputs that fire in round t. For any round with $k_t \geq 2$, we have $pot(z_s, t) \geq 2\,\mathrm{w^{out}} - b(z_s) = 1.5$ and $pot(z_c, t) \geq 2\,\mathrm{w^{out}} - b(z_c) = 1$. So both inhibitors fire in round t w.h.p. Conditioning on this event, all active outputs have:

$$pot(y_j, t+1) = (1 \cdot \mathrm{w^{inh}}_s) + (1 \cdot \mathrm{w^{inh}}_\ell) + (1 \cdot \mathrm{w^{self}}) + \mathrm{w^{input}} - \mathrm{b^{out}} = -1 - 1 + 2 + 3 - 3 = 0$$

and so fire with probability $1/2$ in round $t+1$. All inactive outputs, which did not fire in round t, do not have an active self loop and hence have $pot(y_j, t) = -2$ and don't fire in round $t+1$ w.h.p. (as discussed, all outputs with $x_j = 0$ also do not fire w.h.p.)

Conditioning on this event, with probability $1/2$, $k_{t+1} \leq k_t/2$. Further,

$$\Pr[k_{t+1} = 0] = 1/2^{k_t} \text{ and } \Pr[k_{t+1} = 1] = k_t \cdot (1/2^{k_t}) \geq \Pr[k_{t+1} = 0].$$

So the probability of reaching $k_{t+1} = 1$ and hence N converging to WTA is at least as high as the probability of overshooting WTA and having no outputs firing in round $t+1$.

Conditioning on the fact that z_s and z_c fire in every round in which $k_t \geq 2$ and that no output which was inactive in round t fires in round $t+1$, whenever $k_t \geq 2$ it decreases by a factor of $1/2$ in round $t+1$ with good probability. So w.h.p. within $O(\log(k_0)) = O(\log n)$ rounds there is a round t with either $k_t = 1$ or $k_t = 0$. $k_t = 1$ is at least as likely as $k_t = 0$ so with constant probability, N converges to WTA within $O(\log n)$ rounds. ◀

Two Inhibitor Lower Bound

▶ **Theorem 15.** *For any basic WTA network* N *with* $\alpha = 2$ *inhibitors,* $\mathcal{ET}(\mathrm{N}) = \Omega(\log n / \log\log n)$ *and* $\mathcal{HT}(\mathrm{N}) = \Omega(\log^2 n / \log\log^2 n)$.

The key idea is that the use of a stability inhibitor z_s and a convergence inhibitor z_c in the algorithm is not just a design choice, but is required for *any* near-optimal two inhibitor WTA network.

▶ **Claim 16.** *For any basic WTA network* N *with* $\alpha = 2$ *inhibitors and* $\mathcal{ET}(\mathrm{N}) = O(\log^3 n)$*, one inhibitor* z_s *fires w.h.p. in sub-round* $(t, 3)$ *if at least one output fires in sub-round* $(t, 2)$*. The second inhibitor* z_c*, does not fire w.h.p. in* $(t, 3)$ *if just a single output fires in* $(t, 2)$*.*

Proof. Assume for contradiction that both inhibitors fire with probability $\omega(1/n^c)$ in sub-round $(t, 3)$ after just a single output fires in sub-round $(t, 2)$. Then, after a round t in which $z_s^t = z_c^t = 1$, any output y_j with $x_j = 1$ and $y_j^t = 1$ must fire w.h.p. in round $t+1$. This is because once N converges to WTA, when the single winning output fires in sub-round $(t, 2)$, by our assumption, with relatively high $\omega(1/n^3)$ probability, both z_s and z_c fire in sub-round $(t, 3)$. Even if this event occurs, the winning output must fire w.h.p. in round $t+1$ to maintain WTA w.h.p.

However, if we let $\mathrm{X} = \vec{1}$ and $\mathrm{Y}^0 = \vec{1}$, then for some constant c_1, all outputs will continue firing for $\omega(n^{c_1})$ rounds w.h.p. even if both z_s and z_c fire in every round. This contradicts our assumed $O(\log^3 n)$ runtime. Hence we have that at least one of the inhibitors, which we label z_c, fires with probability $O(1/n^c)$ in sub-round $(t, 3)$ if just a single output fires in sub-round $(t, 2)$.

Similarly, assume for contradiction that z_s *does not fire* with probability $\omega(1/n^c)$ in sub-round $(t, 3)$ if a single output fires in sub-round $(t, 2)$. Then, it must be that even if neither inhibitor fires in sub-round $(t, 3)$, any output y_j that did not fire in sub-round $(t, 2)$ (i.e. $y^t = 0$), must also not fire w.h.p. in sub-round $(t+1, 2)$. This is because, by our assumption, after WTA is reached, with probability $(1 - O(n^c)) \cdot \omega(1/n^c) = \omega(1/n^c)$ neither inhibitor will fire in sub-round $(t, 3)$ when just the single winning output fires in sub-round $(t, 2)$. Still, all non-winning outputs must continue not firing in round $t+1$ to maintain WTA w.h.p.

However, if we let $\mathrm{X} = \vec{1}$ and $\mathrm{Y}^0 = \vec{0}$, since even when neither inhibitor fires in round t, each output does not fire in round $t+1$ w.h.p. if it did not fire in round t, it will take $\omega(n^{c_1})$ rounds (for some constant c_1) before even a single output fires w.h.p. contradicting our assumed $O(\log^3 n)$ runtime. ◀

The above claim allows us to strongly constrain the behavior of the network based on the action of the inhibitors z_s and z_c. Let p_0 be the probability that an output y_j fires in round $t+1$ given that $y^t = 0$, $x^t = 1$ and $z_s^t = z_c^t = 0$.

▶ **Claim 17.** *For any basic WTA network* N *with* $\alpha = 2$ *inhibitors and* $\mathcal{ET}(\mathrm{N}) = o(\log^2 n)$, $p_0 = \omega(1/\log^2 n)$.

Proof. Consider X with just two firing inputs $x_1 = 1$ and $x_2 = 1$. For any round t in which $y_1^t = y_2^t = 0$, the probability that y_1 or y_2 fires in round $t+1$ is *at most* p_0 – since the firing of z_s or z_c can only decrease the probability of the outputs firing. Assuming by way of contradiction that a $p_0 \leq c_1/\log^2 n$ for some constant c_1, starting from $\mathrm{Y}^0 = \vec{0}$, with constant probability, neither output will fire for $\Omega(\log^2 n)$ consecutive rounds, and so N cannot converge to WTA in expected $o(\log^2 n)$ rounds. ◀

Let p_{out} be the probability that output y_j fires in round $t+1$ given $y^t = 1$, $x^t = 1$ and $z_s^t = z_c^t = 1$.

▶ **Claim 18.** *For any basic WTA network* N *with* $\alpha = 2$ *inhibitors and* $\mathcal{ET}(\mathrm{N}) = o(\log^2 n)$, $p_{out} = \omega(1/\log^6 n)$.

Proof. Consider X with $\Theta(\log^4 n)$ firing inputs and initial configuration Y^0 where $y_j = 1$ for all j with $x_j = 1$. Consider some round t in which at least two outputs (corresponding to firing inputs) have fired in all rounds $t' \leq t$. If either (or both) of z_s or z_c do not fire in round t, then since they face at most as much inhibition as when the network has converged to WTA, all outputs with firing inputs that fired in round t fire w.h.p. in round $t+1$. However, if both z_s and z_c do fire in round t, if $p_{out} = O(1/\log^6 n)$ then with probability $\leq (1-p_{out})^{\Theta(\log^4 n)} = 1 - \Theta(1/\log^2 n)$ *no* output corresponding to a firing input fires in round $t+1$. Since by Claim 16 a single inhibitor firing is enough to maintain convergence to WTA, once these outputs do not fire in some round t, they do not fire again w.h.p. until a round in which neither z_s or z_c fire. Then by Claim 17 and a Chernoff bound (Theorem 12) $\omega(\log^2 n)$ of them fire w.h.p.

So overall, we alternate between having many (between $\omega(\log^2 n)$ and $\Theta(\log^4 n)$) outputs corresponding to firing inputs and 0 outputs with firing inputs. Each time we have many firing outputs, with probability at least $1 - \Theta(\log^2 n)$ we have no firing outputs in the next round. So it takes at least $\Omega(\log^2 n)$ rounds before we have a round with exactly one valid firing output with constant probability, contradicting our assumed runtime of $\mathcal{ET}(\mathrm{N}) = o(\log^2 n)$. ◀

With the above claims in place, we are ready to prove Theorem 15. Consider $\mathrm{X} = \vec{1}$ and initial configuration with $\mathrm{Y}^0 = \vec{1}$. Let $k_t = \|Y^t\|_1$ be the number of outputs that fire in round t. Now, if y_j fires in round t, then it fires with probability *at least* p_{out} in round $t+1$, since p_{out} is the firing probability with maximum inhibition. Let $d = c_1 \log n/p_{out}$ for some constant c_1. By Claim 18, $d = O(\log^7 n)$ and since $p_{out} \leq 1$, trivially $d = \Omega(\log n)$. Starting from Y^0 with all outputs firing, for $t = c_2 \frac{\log(d/n)}{\log p_{out}}$ for sufficiently small c_2 we have that any output fires in all rounds up to t with probability $\theta\left(p_{out}^t\right) = \omega\left(\frac{d}{n}\right)$. So by a Chernoff bound (Theorem 12) w.h.p. $\omega(d)$ outputs fire in all rounds $t' \leq t$.

Let t_f represent the first round in which $\leq d$ outputs fire. By our argument above, w.h.p.

$$t_f = \Theta(\log(n/d)/\log(1/p_{out})) = \Theta(\log n/\log(1/p_{out})) = \Omega(\log n/\log\log n) \tag{3}$$

by Claim 18. This gives us $\mathcal{ET}(\mathrm{N}) = \Omega(\log n/\log\log n)$. So it just remains to show our lower bound on $\mathcal{HT}(\mathrm{N})$.

Since $> d$ outputs fire in round $t_f - 1$, again by a Chernoff bound, w.h.p. $k_{t_f} \geq d \cdot p_{out} = \Omega(\log n)$. Consider any round $t > t_f$ in which $k_{t'} > 1$ for all $t' \leq t$. If either of z_s of z_c do not fire in round t, then $k_{t+1} = k_t > 1$ w.h.p. Otherwise, $\Pr[k_{t+1} = 1] = k_t \cdot p_{out}(1 - p_{out})^{k_t - 1}$ and:

$$\Pr[k_{t+1} = 0] = (1 - p_{out})^{k_t} = \Pr[k_{t+1} = 1]] \cdot \frac{1 - p_{out}}{k_t p_{out}} \geq \Pr[k_{t+1} = 1]] \cdot \frac{1}{\log^8 n}$$

where we use the fact that $k_t \leq d = O(\log^7 n)$ and $1 - p_{out} \geq \log n$ or else by (3) we would already not reach WTA w.h.p. in $O(\log^2 n)$ rounds.

So, the probability that $k_{t+1} = 0$ is high (within a polylog n) factor of the probability that $k_{t+1} = 1$. So, with probability at least $\Omega(1/\log^8 n)$, t_f is followed by a reset round in 0 outputs fire before a round in which a single output fires. Further, once such a reset round occurs, then no output will fire until z_s and z_c don't fire in a round (and hence inhibition is lower than it is after convergence to WTA) in which case by Claim 17 $\omega(n/\log^2 n)$ outputs will fire. So w.h.p. there will be $\Omega(\log n/\log\log n)$ rounds before another round in which ≤ 1 outputs fire.

Overall, in order to have a round in which exactly 1 output fires w.h.p. requires $\Omega(\log n/\log(\log^8 n)) = \Omega(\log n/\log\log n)$ resets, each taking $\Omega(\log n/\log\log n)$ rounds, and giving our final lower bound of $\Omega(\log^2 n/\log\log^2 n)$.

B.2 WTA with One Inhibitor

One Inhibitor Lower Bound

▶ **Theorem 19.** *For any basic WTA network* N *with* $\alpha = 1$ *inhibitors,* $\mathcal{ET}(\mathrm{N}) = \Omega(n^c)$.

We fix any constant c and assume by way of contradiction that there is a network N which converges to WTA in $O(n^c)$ rounds in expectation. Let z denote the single inhibitor in N. We first argue that N must be at least somewhat active – given no firing activity from the outputs Y and the inhibitor z, each output connected to an active input should fire with reasonably high probability.

▶ **Claim 20** (Sufficiently Active Network)**.** *If* $z^t = 0$ *then each output* y_j *with* $x_j = 1$ *and* $y_j^t = 0$ *fires in round* $t + 1$ *with probability* $\Omega(1/n^c)$.

Proof. Let X be an input in which exactly one input x_j fires and let $Y^0 = \vec{0}$. The time for N to converge to WTA is lower bounded by the time required for y_j to fire at least once.

Let p_0 be the probability that y_j fires in round $t + 1$ if $y_j^t = 0$ and $z^t = 0$ and let p_1 be the probability that y_j fires in round $t + 1$ if $y_j^t = 0$ and $z^t = 1$. $p_1 \leq p_0$, so as long as y_j does not fire in round t, it fires with probability at most p_0 in round $t + 1$. If $p_0 \leq c_1/n^c$ for some constant c_1 then starting from C_0, with constant probability, y_j will not fire for $\Omega(n^c)$ consecutive rounds. By our assumption that N converges to WTA in $O(n^c)$ rounds in expectation, we have $p_0 = \Omega(1/n^c)$. ◀

We next show that the inhibitor z must fire in round t w.h.p. whenever at least one output fires, in order to maintain stability once WTA has been reached.

▶ **Claim 21** (Stability)**.** *For any configuration* $\mathcal{C}^t$ *of* N*, if at least one output neuron fires in round* t *(i.e.* $\| \mathrm{Y}^t \|_1 \geq 1$*), z fires in round* t *w.h.p.*

Proof. Consider input $X = \vec{1}$. Let t be a round in which WTA is satisfied (exactly one output y_j fires while no other outputs fire). Using the notation of Claim 20, the probability that a non-firing output fires in round $t+1$ is:

$$\Pr[z^t = 1 | Y^t] \cdot p_1 + \Pr[z^t = 0 | Y^t] \cdot p_0.$$

By Claim 20 we have $p_0 \geq c_1/n^c$ for some constant c_1. Since N converges to WTA it must be that w.h.p. in round $t+1$, y_j continues firing and no other output fires. So we have, for some large constant c_2:

$$\begin{aligned} \Pr[z^t = 1 | Y^t] \cdot p_1 + \Pr[z^t = 0 | Y^t] \cdot p_0 &\leq 1/n^{c_2} \\ \Pr[z^t = 0 | Y^t] \cdot c_1/n^c &\leq 1/n^{c_2} \\ \Pr[z^t = 0 | Y^t] &= O(1/n^{c_2 - c}) \end{aligned}$$

which gives the claim as long as $c < c_2$ since exactly one output fires in Y^t. The probability that z fires when > 1 output fires is at least as large due to the excitatory nature of the outputs. ◀

Finally, by way of contradiction, we show that when z fires, any output must stop firing with reasonably high probability. Otherwise, starting with multiple firing outputs, it will take too long to converge to WTA. As we will see this convergence requirement conflicts with the stability requirement of Claim 21 since it means that the winning output will stop firing with reasonably high probability after convergence to WTA.

▶ **Claim 22** (Convergence). *If $z^t = 1$ then y_j with $y_j^t = 1$ and $x_j = 1$ does not fire in round $t+1$ with probability $\Omega(1/n^c)$.*

Proof. Let p denote the probability that an output which corresponds to a firing input and which fires in round t does not fire in round $t+1$ given that $z^t = 1$. We want to show that $p = \Omega(1/n^c)$.

Let $X = \vec{1}$ and let t be any round in which at least two outputs fire. By Claim 21, $z^t = 1$ w.h.p. and at least two outputs fire in round 1 with probability $(1-p)^2 \geq 1 - 2p$. If we start from $Y^0 = \vec{1}$, then w.h.p. at least two outputs will fire in $\Theta\left(\frac{1}{p}\right)$ consecutive rounds. By assumption N converges to WTA within $O(n^c)$ rounds in expectation so we must have $p = \Omega(1/n^c)$. ◀

Putting it all together, consider an execution that satisfies WTA in round t with exactly one output y_j firing. Then, by Claim 21, z fires in round t w.h.p. Thus, by Claim 22, y_j stops firing in round $t+1$ with probability $\Omega(1/n^c)$, in contradiction to the fact that the network must eventually converge to WTA and have y_j fire for n^{c_1} consecutive rounds for some large constant c_1. We briefly note that the above lower bound can be matched with a trivial single inhibitor algorithm.

▶ **Observation 23.** *There is basic network N with $\alpha = 1$ inhibitors with $\mathcal{ET}(N) = O(n^c)$.*

Proof. The single inhibitor z simply fires w.h.p. in round t whenever ≥ 1 outputs fire in round t. The weights are set such that when $z^t = 1$ and $y_j^t = 1$, y_j fires in round $t+1$ with probability $1/n^{c+1}$. If z does not fire, any y_j with $x_j = 1$ fires w.h.p.

It is not hard to see that starting with any input, we will reach a round satisfying WTA within $O(n^c)$ rounds in expectation and after this round is reached, WTA will be maintained for $O(n^{c-1})$ additional rounds in expectation (and so $O(n^{c-2})$ w.h.p.). ◀

B.3 WTA with $O(\log n)$ Inhibitors

Proof of Theorem 4 ($O(\log n)$ Inhibitor Upper Bound). Recall that we assume w.l.o.g. $1/\lambda = c_1 \log n$ for some constant c_1. We set $\mathrm{w}^{\mathrm{input}} = 3$, $\mathrm{w}^{\mathrm{self}} = 2$, and $\mathrm{b}^{\mathrm{out}} = 3$. In this way, exactly as in the two inhibitor network analyzed in Section B.1, any output y_j with $x_j = 0$ will have $pot(y_j, t) \leq \mathrm{w}^{\mathrm{self}} - \mathrm{b}^{\mathrm{out}} = -1$ in every round t and so will not fire w.h.p. in any round.

Our network has $\alpha = \lceil \log n \rceil$ inhibitors. The first is a stability inhibitor z_s, which behaves exactly as the stability inhibitor in the two inhibitor network analyzed in Section B.1. ${\mathrm{w}^{\mathrm{out}}}_s = 1$, $b(z_s) = 0.5$ and ${\mathrm{w}^{\mathrm{inh}}}_s = -1$. z_s fires w.h.p. in sub-round $(t, 3)$ if ≥ 1 output fires in sub-round $(t, 2)$ and does not fire w.h.p. if no output fires. We also have $\alpha - 1$ convergence inhibitors $z_1, ..., z_{\alpha-1}$. For each z_i, $b(z_i) = 2^i - .5$ and ${\mathrm{w}^{\mathrm{out}}}_i = 1$. Therefore, z_i fires w.h.p. in round t whenever $\geq 2^i$ outputs fire in the round. It does not fire w.h.p. if $< 2^i$ outputs fire. We set the inhibitor weight from z_1 to each output to be ${\mathrm{w}^{\mathrm{inh}}}_1 = -1$. For each $i \in 2, ..., \alpha - 1$ we set ${\mathrm{w}^{\mathrm{inh}}}_i = -\lambda \cdot \log_2(e)$.

We can see that the stability Claim 13 holds just as it does in the two inhibitor network analyzed in Section B.1. Specifically, if just a single output y_j with $x_j = 1$ fires in some round t, w.h.p. z_s will fire while the convergence inhibitors will all not fire. So we will have:

$$pot(y_j, t+1) = {\mathrm{w}^{\mathrm{inh}}}_s + \mathrm{w}^{\mathrm{input}} + 1 \cdot \mathrm{w}^{\mathrm{self}} - \mathrm{b}^{\mathrm{out}} = -1 + 3 + 2 - 3 = 1$$

so y_j fires w.h.p. in round t+1. At the same time for $j' \neq j$, since $y_{j'}$ does not fire in round t:

$$pot(y_{j'}, t+1) = {\mathrm{w}^{\mathrm{inh}}}_s + \mathrm{w}^{\mathrm{input}} + 0 \cdot \mathrm{w}^{\mathrm{self}} - \mathrm{b}^{\mathrm{out}} = -1 + 3 + 0 - 3 = -1$$

so $y_{j'}$ will not fire in round $t+1$. So, once a single y_j with $x_j = 1$ fires in some round t, N will converge to WTA w.h.p. We now show that N reaches such a round in $O(1)$ expected time.

Consider any round $t > 0$ in which $k_t \geq 2$ outputs fire. We can assume that all these outputs corresponding to firing inputs since as discussed, outputs corresponding to non-firing inputs do not fire w.h.p. in any round. For some i we have $k_t \in [2^i, 2^{i+1})$ and so w.h.p. in round t, $z_s, z_1, ..., z_i$ fire while all other inhibitors do not fire (note that $\alpha - 1 = \lceil \log n \rceil - 1$ and so even if n outputs fire, all inhibitors fire). We thus have, w.h.p. for any active output y_j with $y_j^t = 1$ and $x_j = 1$:

$$\begin{aligned} pot(y_j, t+1) &= \mathrm{w}^{\mathrm{self}} + \mathrm{w}^{\mathrm{input}} - \mathrm{b}^{\mathrm{out}} + {\mathrm{w}^{\mathrm{inh}}}_s + {\mathrm{w}^{\mathrm{inh}}}_1 + \sum_{j=2}^{i} {\mathrm{w}^{\mathrm{inh}}}_j \\ &= 2 + 3 - 3 - 1 - 1 - (i-1)\lambda = (i-1)\lambda \cdot \log_2(e). \end{aligned}$$

So y_j fires in round $t+1$ with probability:

$$p(y_j, t+1) = \frac{1}{1 + e^{(i-1)\lambda \log_2(e)/\lambda}} = \frac{1}{1 + 2^{i-1}}$$

Since $k_t \in [2^i, 2^{i+1})$, we have $1 \leq \frac{k_t}{1+2^{i-1}} \leq 4$ and so can bound the probability that exactly one output that was active in round t fires in round $t+1$ as:

$$\begin{aligned} k_t \cdot \frac{1}{1+2^{i-1}} \cdot \left(1 - \frac{1}{1+2^{i-1}}\right)^{k_t - 1} &\geq \left(1 - \frac{1}{1+2^{i-1}}\right)^{k_t - 1} \\ &\geq \left(1 - \frac{1}{1+2^{i-1}}\right)^{4(1+2^{i-1})} \\ &\geq \frac{1}{4^4}. \end{aligned}$$

So, with constant probability exactly one output that fired in round t also fires in round $t+1$. Any output that did not fire in round t has potential $\leq \mathrm{w}^{\mathrm{input}} - \mathrm{b}^{\mathrm{out}} + \mathrm{w}^{\mathrm{inh}}{}_s + \mathrm{w}^{\mathrm{inh}}{}_\ell = -2$ and so does not fire with high probability. So, with constant probability, exactly one output y_j with $x_j = 1$ fires, and so N converges to WTA.

We conclude by noting that, by the arguments of Claim 14 for our two inhibitor network, with constant probability, starting with any Y^0 we in fact have a round with $k_t \geq 1$ firing outputs all with active inputs within constant rounds. So from any starting configuration, we converge to WTA with constant probability in $O(1)$ rounds. Repeating this constant probability argument gives both $\mathcal{ET}(\mathrm{N}) = O(1)$ and $\mathcal{HT}(\mathrm{N}) = O(\log n)$. ◀

$\Omega(\log n)$ High Probability Runtime Lower Bound

▶ **Theorem 24.** *Any basic WTA network* N*, with any number of inhibitors, has* $\mathcal{HT}(\mathrm{N}) = \Omega(\log n)$*.*

Proof. We show that any network N requires $\Omega(\log n)$ rounds before a round t in which WTA is satisfied w.h.p. This immediately gives our lower bound on convergence time.

Consider input $X = \vec{1}$ (so any output is a valid winner) and any round t such that WTA has not been satisfied for any $t' < t$. That is, in no round t' does exactly one output y_j fire. Let W_t be the event that in round t exactly one output fires and hence WTA is satisfied. We claim that $\Pr[W_t = 1 \mid C^{t-1}] \leq c$ for any configuration C^{t-1} of N in round $t-1$ and some universal constant c. That is, no matter the network configuration in round $t-1$, WTA will only be achieved with constant probability in the next round. Hence, as long as the initial output configuration Y^0 is one in which WTA is not satisfied, for $t = O(\log n)$, with probability at least $(1-c)^t = \Omega(1/n^{c'})$, for some constant c', WTA will not be satisfied in any even round up to t. This gives that $\mathcal{HT}(\mathrm{N}) = \Omega(\log n)$. There are two cases:

Network Reset

$Y^{t-1} = \vec{0}$. In this case, no output fired in round $t-1$. Since all outputs are identical w.r.t their edge weights and bias values, conditioned on the behavior Z^{t-1} of the inhibitors in round $t-1$, all outputs will fire independently with some fixed probability p in round t. For any p and any $n \geq 2$, the probability that exactly 1 will fire in round t is:

$$\Pr[W_t = 1 \mid C^{t-1}] = n \cdot p(1-p)^{n-1} \leq \frac{1}{2}.$$

No Reset

$\|Y^{t-1}\|_1 \geq 2$ – i.e. there are at least 2 firing outputs in round $t-1$. Let O_1 be the set of firing outputs in round $t-1$ and O_0 be the set of non-firing outputs. Conditioned on Z^{t-1}, any output in O_1 fires independently with some probability p_1 in round t and any output in O_0 fires with some probability p_0. Further, $p_0 \leq p_1$ since the only difference in membrane potential between the neurons in O_0 and O_1 will be whether their excitatory self loop is active.

For $a \in \{0,1\}$ let V_a be the event that exactly 1 output from O_a fires in round t. Clearly, $W_t \subseteq V_1 \cup V_0$. For any p_1, $\Pr[V_1 \mid C^{t-1}] = |O_1| \cdot p_1(1-p_1)^{|O_1|-1} \leq 1/2$ since we have not reached WTA and so $|O_1| \geq 2$. If $|O_0| = 0$, then vacuously, $\Pr[V_0 \mid C^{t-1}] = 0$ and hence $\Pr[W_t \mid C^{t-1}] \leq 1/2$. Alternatively, If $|O_0| \geq 2$ then we also have $\Pr[V_0 \mid C^{t-1}] \leq 1/2$ and, since all outputs fire independently conditioned on C^{t-1},

$$\Pr[W_t \mid C^{t-1}] \leq 1 - \Pr[\neg(V_1 \cup V_0)] \leq 1 - (1-1/2)^2 = 3/4.$$

Finally, if $|O_0| = 1$ either $p_0 \leq 1/2$, in which case $\Pr[V_0 \mid C^{t-1}] \leq 1/2$ and we again have $\Pr[W_t \mid C^{t-1}] \leq 3/4$ or $p_0 \geq 1/2$ in which case $p_1 \geq p_0 \geq 1/2$, and the probability that at least two outputs from O_1 fire is at least $1/4$ and hence WTA is achieved with probability at most $3/4$. ◀

B.4 WTA with $\alpha \geq 2$ Inhibitors

Proof of Theorem 5. We first describe the network construction in detail. As in our previous networks, we have a stability inhibitor z_s that fires w.h.p. whenever ≥ 1 outputs fire in round t. This inhibitor ensures that in round $t+1$ w.h.p. only outputs that fired in round t (and hence have an active self loop) will fire in round $t+1$.

We set the excitatory input to output connection weight to $\mathrm{w}^{\mathrm{input}} = 3$, the excitatory output self-loop to $\mathrm{w}^{\mathrm{self}} = 2$, and the output bias to $\mathrm{b}^{\mathrm{out}} = 3$. For the stability inhibitor we set the excitatory output to inhibitor weight ${\mathrm{w}^{\mathrm{out}}}_s 1$, $b(z_s) = .5$, and ${\mathrm{w}^{\mathrm{inh}}}_s = -1$ just as we did in the two inhibitor algorithm.

We have θ groups each containing $\lceil (\log n)^{1/\theta} \rceil$ convergence inhibitors, $Z_1, Z_2, ..., Z_\theta$ where we denote $Z_i = \{z_{i,1}, z_{i,2}, ..., z_{i,\lceil (\log n)^{1/\theta} \rceil}\}$. We set ${\mathrm{w}^{\mathrm{out}}}_i = 1$ for all $i \in Z_1, Z_2, ..., Z_\theta$ and $b(z_{i,j}) = 2^{jd_i} - .5$. In this way, when $k_t \in \left[2^{jd_i}, 2^{(j+1)d_i}\right)$ w.h.p. $z_s, Z_1, ..., Z_{i-1}, z_{i,1}, ..., z_{i,j}$ all fire while the remaining inhibitors do not. We set ${\mathrm{w}^{\mathrm{inh}}}_{i,j}$ such that

$$pot_{i,j} = \mathrm{w}^{\mathrm{input}} + \mathrm{w}^{\mathrm{self}} + {\mathrm{w}^{\mathrm{inh}}}_s - \mathrm{b}^{\mathrm{out}} + \sum_{\{(k,l) | k<i \text{ or } l \leq j\}} {\mathrm{w}^{\mathrm{inh}}}_{k,l}$$

satisfies:

$$p_{i,j} = \frac{1}{1 + e^{-pot_{i,j}/\lambda}} = \frac{c_1}{2^{jd_i}}$$

for some small constant c_1. For simplicity of presentation, we do not explicitly calculate out these weights. However, it is clear that choosing correct weights $p_{i,j}$ decreases as most inhibitors fire and the sigmoid function is continuous and decreases monotonically as pot_i decreases. We are now ready to analyze the network behavior in detail.

No Firing Inputs

As in the two inhibitor network, any y_j with $x_j = 0$, has maximum potential is $\mathrm{w}^{\mathrm{self}} - \mathrm{b}^{\mathrm{out}} = -1$ (even when no inhibitors fire) so and will not fire w.h.p. outside of the initial configuration Y^0. ($p(y_j, t) \leq \frac{1}{1+e^{1/\lambda}} \leq 1/n^c$ for any t since $\lambda = 1/c_1 \log n$). If $\mathrm{X} = \vec{0}$, this implies that a valid WTA state in which no outputs fire will be converged to w.h.p. trivially. We now focus on the case when $\| \mathrm{X} \|_1 \geq 1$.

Maintaining WTA (Stability)

If just a single output y_j corresponding to an active input ($x_j = 1$) fires in round t then w.h.p. by Claim 13 in Appendix B.1, N converges to WTA. This is because w.h.p. just z_s will fire in round t and y_j has potential

$$pot(y_j, t+1) = (1 \cdot {\mathrm{w}^{\mathrm{inh}}}_s) + (0 \cdot {\mathrm{w}^{\mathrm{inh}}}_\ell) + (1 \cdot \mathrm{w}^{\mathrm{self}}) + \mathrm{w}^{\mathrm{input}} - \mathrm{b}^{\mathrm{out}} = -1 + 2 + 3 - 3 = 1.$$

So y_j fires with probability $\frac{1}{1+e^{c_1 \log n}} \geq 1 - 1/n^c$ in round $t+1$. In contrast, for any $j' \neq j$, $y_{j'}$ does not fire in round t so has

$$pot(y_{j'}, t+1) \leq (1 \cdot {\mathrm{w}^{\mathrm{inh}}}_s) + (0 \cdot {\mathrm{w}^{\mathrm{inh}}}_\ell) + (0 \cdot \mathrm{w}^{\mathrm{self}}) + \mathrm{w}^{\mathrm{input}} - \mathrm{b}^{\mathrm{out}} = -1 + 3 - 3 = -1.$$

Therefore y'_j fires with probability $\leq 1/n^c$ in round $t+1$ so WTA is satisfied with output y_j firing in round $t+1$ w.h.p.

Converging to WTA

It now just remains to show that with constant probability, within $O(\theta)$ rounds, there is at least one round in which exactly one output y_j with $x_j^t = 1$ fires. By the stability argument above once such a round occurs, N will converge to WTA w.h.p.

By the arguments of the convergence Claim 14 for the two inhibitor network, with constant probability, starting with any Y^0 we in fact have a round with $k_t \geq 1$ firing outputs all with active inputs within constant rounds. If $k_t = 1$ then N converges to WTA and we are done. So it suffices to consider the case when $k_t \geq 2$.

If $k_t \in \left[2^{jd_i}, 2^{(j+1)d_i}\right)$ then w.h.p. $z_s, Z_1, ..., Z_{i-1}, z_{i,1}, ..., z_{i,j}$ fire while the other inhibitors do not and so in round $t+1$ any active output that fired in round t fires with probability $p_{i,j}$. So we have $E[k_{t+1}] \in [1, c_1 2^{d_i})$, and, so with at least constant probability by a Markov bound $k_{t+1} < 2^{d_i}$ if we set c_1 to a small constant.

Additionally, in any round with $k_t \geq 2$ conditioning on the high probability event that the correct inhibitors fire,

$$\Pr[k_{t+1} = 1] = k_t \cdot p_{i,j}(1 - p_{i,j})^{k_t - 1}$$

and:

$$\begin{aligned}
\Pr[k_{t+1} = 0] = (1 - p_{i,j})^{k_t} &= \Pr[k_{t+1} = 1] \cdot \frac{(1 - p_{i,j})}{k_t p_{i,j}} \\
&\leq \Pr[k_{t+1} = 1] \cdot \frac{1}{2^{jd_i} \cdot c1/2^{jd_i}} \\
&\leq \frac{1}{c_1} \Pr[k_{t+1} = 1].
\end{aligned}$$

So, the probability of having exactly one output fire and hence converging to WTA is within a constant factor of the probability or having 0 outputs fire and 'reseting' the network. So overall with constant probability, we reach such a round with $k_t = 1$ within just $O(\theta)$ rounds. Iterating this argument gives the expected and high probability runtime bounds of Theorem 5. ◀

Proof of Theorem 6. Again we have a stability inhibitor z_s that fires w.h.p. in sub-round $(t, 3)$ whenever ≥ 1 outputs fire in sub-round $(t, 2)$. We also have a 'base level' convergence inhibitor that fires w.h.p. whenever ≥ 2 outputs fire. When just z_s and z_ℓ fire in round t, any output (with an active input) that fired in round t fires with probability $1/2$ in round $t+1$.

We then employ $\alpha - 2$ additional convergence inhibitors $z_1, ...z_{\alpha-2}$. For $i \in 1, ..., \alpha - 2$ let

$$d_i = (\log n)^{i/(\alpha-1)}.$$

Letting k_t be the number of outputs that fire in round t, z_i fires w.h.p. in round t whenever $k_t \geq 2^{d_i}$. The synapse weights from the inhibitors to the outputs are chosen such that, when $k_t \in \left[2^{d_i}, 2^{d_{i+1}}\right)$, and hence $z_1, ..., z_i$ each active output (i.e. each y_j with $y_j^t = 1$ and $x_j = 1$) fires with probability:

$$p_i = \frac{c \log n}{d_i} = \frac{c \log n}{(\log n)^{i/(\alpha-1)}}$$

in round $t+1$. This probability is enough to ensure that within few rounds, we will have $< 2^{d_i}$ active outputs. Specifically, since $k_t \in \left[2^{d_i}, 2^{d_{i+1}}\right)$, for

$$r = \frac{\log k_t}{\log 1/p_i} \leq \frac{(\log n)^{(i+1)/(\alpha-1)}}{(\log n)^{i/(\alpha-1)} - \log(c \log n)} = O\left((\log n)^{1/(\alpha-1)}\right)$$

with high probability, there will be a round $r' = O(r)$ with $k_{t+r'} \leq 2^{d_i}$. At the same time, p_i is large enough that w.h.p. we will not overshoot WTA and have 0 firing outputs in round $t + r'$. Even if $k_t = 2^{d_i}$ then we have $k_t \cdot p_i = c \log n$ and so, for large enough c, with high probability, by a Chernoff bound (Theorem 12) at least $O(\log n)$ outputs fire in round $t+1$.

Overall, within $O\left((\alpha-2)(\log n)^{1/(\alpha-1)}\right)$ rounds, the number of active outputs falls within $[2, 2^{d_1}]$ w.h.p. Once k_t is in this range, just z_s and z_1 fire w.h.p. so our network is essentially identical to the two inhibitor network described in the previous section and analyzed in detail in Appendix B.1. We thus reach WTA with constant probability in $\Theta(\log 2^{d_1}) = \Theta\left((\log n)^{1/(\alpha-1)}\right)$ additional rounds, giving our final runtime bound of $O\left(\alpha(\log n)^{1/(\alpha-1)}\right)$.

We now formalize the above arguments. Following our earlier constructions, we set the excitatory input to output connection weight to $\mathrm{w}^{\mathrm{input}} = 3$, the excitatory output self-loop to $\mathrm{w}^{\mathrm{self}} = 2$, and the output bias to $\mathrm{b}^{\mathrm{out}} = 3$. Set the excitatory output to inhibitor weights ${\mathrm{w}^{\mathrm{out}}}_s = {\mathrm{w}^{\mathrm{out}}}_\ell = 1$, $b(z_s) = .5$, $b(z_\ell) = 1.5$, and ${\mathrm{w}^{\mathrm{inh}}}_\ell = {\mathrm{w}^{\mathrm{inh}}}_s = -1$ just as we did in the two inhibitor algorithm.

For the additional convergence inhibitors, set ${\mathrm{w}^{\mathrm{out}}}_i = 1$ for all $i \in 1, ..., \alpha - 2$ and $b(z_i) = 2^{d_i} - .5$. In this way, when $k_t < 2^{d_1}$, w.h.p. just z_s and z_1 fire, and each active output in round t has potential

$$pot(y_j, t+1) = \mathrm{w}^{\mathrm{input}} + \mathrm{w}^{\mathrm{self}} + {\mathrm{w}^{\mathrm{inh}}}_s + {\mathrm{w}^{\mathrm{inh}}}_\ell - \mathrm{b}^{\mathrm{out}} = 3 + 2 - 1 - 1 - 3 = 0$$

and so fires with probability $p_1 = 1/2$ in round $t+1$. We set ${\mathrm{w}^{\mathrm{inh}}}_i$ such that

$$pot_i = \mathrm{w}^{\mathrm{input}} + \mathrm{w}^{\mathrm{self}} + {\mathrm{w}^{\mathrm{inh}}}_s + {\mathrm{w}^{\mathrm{inh}}}_\ell - \mathrm{b}^{\mathrm{out}} + \sum_{j=1}^{i} {\mathrm{w}^{\mathrm{inh}}}_j$$

satisfies:

$$p_i = \frac{1}{1 + e^{-pot_i/\lambda}} = \frac{c \log n}{2^{d_i}}.$$

As in the proof of Theorem 5, we do not explicitly calculate out these weights. Roughly, ${\mathrm{w}^{\mathrm{inh}}}_i \approx \Theta(\frac{\lambda \log\log n}{\alpha - 1})$ such that when i inhibitors fire $p_i \approx \frac{1}{e^{-\Theta(\frac{i\lambda \log\log n}{\alpha-1})}} \approx \frac{c \log n}{2^{d_i}}$. It is clear that choosing correct weights is possible as $1/2 > p_1 > ... > p_{\alpha-1}$ and the sigmoid function is continuous and decreases monotonically as pot_i decreases.

By identical arguments to those in the proof of Theorem 5, we converge to WTA in constant rounds w.h.p. if there are no firing inputs or if a single output with a firing input fires in a round. Hence it just remains to show that with constant probability, within $O(\alpha(\log n)^{1/(\alpha-1)})$ rounds, there is at least one round in which exactly one output y_j with $x_j^t = 1$ fires.

Again, by the arguments of the convergence Claim 14 for the two inhibitor network, with constant probability, starting with any Y^0 we in fact have a round with $k_t \geq 1$ firing outputs all with active inputs within constant rounds. If $k_t = 1$ then N converges to WTA and we are done. So it suffices to consider the case when $k_t \geq 2$. In this case, as discussed if $k_t \in \left[2, 2^{d_1}\right)$ then w.h.p. just z_s and z_ℓ fire, and so each active output has potential

$$pot(y_j, t+1) = (1 \cdot {\mathrm{w}^{\mathrm{inh}}}_s) + (1 \cdot {\mathrm{w}^{\mathrm{inh}}}_\ell) + (1 \cdot \mathrm{w}^{\mathrm{self}}) + \mathrm{w}^{\mathrm{input}} - \mathrm{b}^{\mathrm{out}} = -1 - 1 + 2 + 3 - 3 = 0$$

and fires with probability $1/2$ in round $t+1$. All inactive outputs, which did not fire in round t, do not have an active self loop and hence have $pot(y_j,t) = -2$ and don't fire in round $t+1$ w.h.p. (as discussed, all outputs with $x_j = 0$ also do not fire w.h.p.)

Conditioning on this event, with probability $1/2$, $k_{t+1} \leq k_t/2$ and by the arguments in Claim 14, we converge to WTA with constant probability within $O(k_t) = O(d_1) = O\left((\log n)^{1/(\alpha-1)}\right)$ rounds.

If $k_t \in \left[2^{d_i}, 2^{d_{i+1}}\right)$ for some $i \in 1, ..., \alpha-2$ then as discussed, w.h.p. $z_s, z_\ell, z_1, ..., z_i$ all fire in round t while all other inhibitors do not fire. We thus have

$$\mathrm{E}[k_{t+1}] \geq 2^{d_i} \cdot p_i = \frac{2^{(\log n)^{i/(\alpha-1)}} \cdot c \log n}{2^{(\log n)^{i/(\alpha-1)}}} = c \log n$$

By a Chernoff bound (Theorem 12), w.h.p. k_{t+1} falls within a constant multiplicative factor of its expection. Thus, w.h.p. we still have $k_{t+1} \geq 2$. At the same time, w.h.p. $k_{t+1} \leq c_1 k_t \cdot p_i$ for some constant c_1. So overall, within $r = \frac{\log k_t}{\log 1/p_i} = O\left((\log n)^{1/(\alpha-1)}\right)$ rounds, w.h.p. $k_{t+r} < 2^{d_i}$. Within $\alpha-2$ epochs of $O\left((\log n)^{1/(\alpha-1)}\right)$ rounds we thus have $k_t \in \left[2, 2^{d_1}\right)$ w.h.p. and then reach WTA withing $O\left((\log n)^{1/(\alpha-1)}\right)$ additional rounds with constant probability.

Iterating this constant proability argument gives the expected and high probability runtime bounds of Theorem 6. ◀

B.5 Missing Proofs for Main Lower Bound (Theorem 7)

B.5.1 Inhibitors are Nearly Deterministic for Most Density Classes

Proof of Lemma 8. By the definition of the set S, for $z \in S$ it holds that z fires in sub-round $(t,3)$ with probability $1/(1+e^{-pot_1(z)}) \geq 1/\log^{3c} n$ and hence $\mathrm{w^{out}}_z - b(z) \geq -3c \log\log n$. By our no-background noise assumption that neurons do not fire w.h.p. with no external input, we can assume $b(z) \geq 3\log n$ and hence have $pot_2(z) = 2\,\mathrm{w^{out}}_z - b(z) \geq 2\log n$. Thus, z fires with probability at least $1 - 1/n^2$ in sub-round $(t,3)$. Overall, all the $|S| \leq O(\log n)$ inhibitors fire in sub-round $(t,3)$, with probability at least $1-1/n$ as required. ◀

Proof of Lemma 9. In any round t, even if all n outputs fire in sub-round $(t,2)$, the firing probability of each inhibitor in R in sub-round $(t,3)$ is at most $1/\log^c n$ (or else the inhibitor would fall in C). Union bounding over the first $O(\log\log n)$ rounds of execution and the at most $O(\log n)$ inhibitors in R, we get that with probability at least $1 - 1/\log^{c-3} n$, none of these inhibitors fires in these rounds. ◀

Proof of Lemma 10. Let $k(z)$ be the smallest integer in $[1,n]$ such that z fires in sub-round $(t,3)$ with probability at least $1/\log^c n$ when $k(z)$ outputs fire in sub-round $(t,2)$. By the definition of C, when n outputs fire, z fires in the next sub-round with probability at least $1/\log^c n$, and hence $k(z)$ is well defined. In addition, since $z \notin S$, $k(z) \geq 2$.

Part (1) of the claim follows immediately by the definition of $k(z)$. To prove part (b), the key idea is to exploit the following gap in the behavior of $z \in C$: since z is not in S, the firing probability of z in steady state (with exactly one firing output) is *at most* $1/\log^{3c} n$. On the other hand, when there are at least $k(z) \geq 2$ active outputs, the firing probability of z is *at least* $1/\log^c n$. This implies that the sigmoid function which converts the number of firing inputs to z's firing probability must be steep enough such that z fires with good probability when $\geq 2k(z)$ outputs fire. By the fact that $z \notin S$, $pot_1(z) = \mathrm{w^{out}}_z - b(z) \leq -3c \cdot \log\log n$

and so $\mathrm{w^{out}}_z \leq b(z) - 3c\log\log n$. On the other hand, by the definition of $k(z)$, $\mathrm{w^{out}}_z$ cannot be too small since $pot_{k(z)}(z) = k(z)\cdot \mathrm{w^{out}} - b(z) \geq -c\cdot\log\log n$ so

$$k(z)\cdot \mathrm{w^{out}}_z \geq b(z) - c\log\log n. \tag{4}$$

Combining this we get: $k(z)b(z) - 3k(z)\cdot c\log\log n \geq b(z) - c\log\log n$ and so $b(z) \geq 3c\log\log n$. Using that and Eq. (4), we get: $pot_{2k(z)}(z) = 2k(z)\cdot \mathrm{w^{out}} - b(z) \geq 2b(z) - 2c\log\log n - b(z) = b(z) - 2c\log\log n \geq c\log\log n$. Hence, $1/(1+e^{-pot_{2k(z)}(z)}) \geq 1 - 1/(\log^c n)$ as required. ◀

B.5.2 Detailed Description of the Prediction Process

In this section we describe the prediction process in more detail.

Inductive Assumptions

For each round t, in showing that we are able to predict the behavior of N for a large number of inputs in round t, we make several inductive assumptions:

For two ranges of positive numbers $R_1 = [r_1, r_2]$ and $R_2 = [r_3, r_4]$ such that $r_1 \leq r_2 \leq r_3 \leq r_4$, and a positive number a, the ranges are called a-separated if $r_3/r_2 \geq a$. The *value* of the range $R_1 = [r_1, r_2]$ is taken to be r_1. We assume that for $\mathrm{X} \in \mathcal{X}_{t-1} \subset \mathcal{X}$ the ranges $R_{t-1}(\mathrm{X})$ are all a separated for some constant a and have minimum value $\Theta(\log^7 n)$. We also assume that our earlier predictions are accurate: for each $X \in \mathcal{X}_{t-1}$, $\widehat{R}_{t-1}(\mathrm{X}) \in R_{t-1}(\mathrm{X})$ and $\widehat{F}_{t-1}(\mathrm{X}) = F_{t-1}(\mathrm{X})$ with probability at least $1 - \Theta(1/\log n)$. We first show that these assumptions hold for round one:

Predicting the number of firing outputs in sub-round $(1,2)$

Since we consider the initial reset configuration $Y^0 = \vec{0}$ we have $\widehat{R}_0(\mathrm{X}_i) = 0$ for all X_i. Trivially we can set $\mathcal{X}_0 = \mathcal{X}$ – we deterministically know the behavior of all outputs in round 0. By our no-background noise assumption, for every $z \in \mathrm{Z}$, $b(z) = c\log n$, and so w.h.p. $\widehat{F}_0(\mathrm{X}_i) = 0$ for all X_i (no inhibitor fires in the initialization round). Let $\mathcal{X}_1^{large} = \{\mathrm{X}_i \mid 2^i \geq \log^9 n\}$ (note that $|\mathrm{X}_1^{large}| = \Theta(\log n)$). Let p_0 be the probability that an output fires in sub-round $(t+1,2)$ given that no inhibitor and no output fires in round t (i.e, no output has an active self-loop). Since there are 2^i active *input neurons* in X_i, conditioned on the high probability event that $\widehat{R}_0(\mathrm{X}_i) = 0$ and $\widehat{F}_0(\mathrm{X}_i) = \vec{0}$, the expected number of firing outputs in sub-round $(1,2)$ is $p_0 \cdot \mathrm{X}_i$. It is not hard to show that $p_0 = \Omega(1/\log^2 n)$ and by combining this fact with a Chernoff bound we have:

▶ **Claim 25.** *For every $\mathrm{X}_i \in \mathcal{X}_1^{large}$, w.h.p. the number of firing outputs in sub-round $(1,2)$, $\widehat{R}_1(\mathrm{X}_i)$ is in the range $R_1(\mathrm{X}_i) = [(1 - 1/\log^3 n)\cdot p_0 2^i, (1 + 1/\log^3 n)\cdot p_0 2^i]$. Hence, the predicted output ranges for the inputs in $\mathcal{X}_1^{large}$ are $2(1 - 1/\log n)$ separated. Additionally each has minimum value $\Omega(\log^7 n)$.*

Proof. Let X_1 be a vector with exactly one firing input and let y_i be its corresponding output. Starting from $Y^0 = \vec{0}$, w.h.p., no inhibitor fires in round 0. If $p_0 < 1/\log^2 n$ then since p_0 rate is the maximum firing probability for y_j in sub-round $(t+1,2)$ given that it didn't fire in sub-round $(t,2)$, the network requires $\Omega(\log^2 n)$ rounds until y_j fires with constant probability and so at least that long to converge to WTA. So we can work in the case where $p_0 \geq 1/\log^2 n$.

For $\mathrm{X}_i \in \mathcal{X}^{large}$ we thus have the expected number of firing outputs in sub-round $(1,1)$ is $p_0 \cdot 2^i \geq 1/\log^2 n \cdot \log^9 n = \log^7 n$. Since the random firings of the outputs are independent

given the firing behavior of the inhibitors and since no inhibitors fire in sub-round $(0,3)$ w.h.p. by a Chernoff bound (Theorem 12), we have that w.h.p. the number of firing outputs $\widehat{R}_1(\mathrm{X}_i)$ is in the range $(1 \pm 1/\log^3 n) \cdot p_0 \cdot 2^i$ for all $\mathrm{X}_i \in \mathcal{X}^{large}$. ◀

The above shows that the predicted ranges for all $\mathrm{X} \in \mathcal{X}_1^{large}$ are well separated, accurate, and have high value. We can now set $\mathcal{X}_1$ to include any $\mathrm{X} \in \mathcal{X}_1^{large}$ except possibly $|C| \leq \alpha$ inputs where $R_1(\mathrm{X})$ overlaps a critical region $K(z)$ for some $z \in C$. Since the remaining ranges do not overlap any critical regions, by Lemmas 8, 9, and 10 we are able to predict $\widehat{F}_1(\mathrm{X})$ with good probability, and so have all our inductive assumptions in round 1.

Predicting the number of firing outputs for $t \geq 2$

We first define a subset of inputs $\mathcal{X}_t^{large} \subseteq \mathcal{X}_{t-1}$ for which we can predict the behavior of the outputs in N in sub-round $(t,2)$. Let $\mathcal{X}_t^{same} \subseteq \mathcal{X}_{t-1}$ be the largest subset of inputs whose predicted firing vector $F_{t-1}(\mathrm{X})$ for the inhibitors in sub-round $(t-1,3)$ is the *same*, and denote this common firing vector by F_{t-1}^*. Let $\mathcal{X}_t^{large}$ be the set of inputs in $\mathcal{X}_t^{same}$ after omitting $\Theta(\log\log n)$ inputs with the smallest range value in sub-round $(t-1,2)$.

Eventually we will show that $\mathcal{X}_t^{large}$ is a reasonably large set of inputs compared to $\mathcal{X}_{t-1}$, and hence we can continue predicting behavior for at least some inputs for a large number of rounds. But first we show how to predict $R_t(\mathrm{X})$ for every input $\mathrm{X} \in \mathcal{X}_t^{large}$.

Let p be the probability that an active output (one with $y_j^{(t-1,2)} = 1$) fires in sub-round $(t,2)$ given that the inhibitors fired in sub-round $(t-1,3)$ according to F_{t-1}^*. Since all inputs in $\mathcal{X}_t^{same}$ have the same predicted firing vector F_{t-1}^*, in each of them, an active output fires in sub-round $(t,2)$ with probability p. In addition, by induction for every $\mathrm{X} \in \mathcal{X}_t^{same} \subseteq \mathcal{X}_{t-1}$, $R_{t-1}(\mathrm{X})$ has a minimum of $\Theta(\log^7 n)$ predicted firing outputs. So inhibition in sub-round $(t-1,3)$ w.h.p. must be at least as high as it is once we have converged to WTA and just a single output is firing. Thus, any output that did not fire in sub-round $(t-1,2)$ must not fire w.h.p. in sub-round $(t,2)$, since non-firing outputs continue not to fire once WTA is converged to.

So just focusing on active outputs that fire in sub-round $(t,2)$, for every $\mathrm{X}_i \in \mathcal{X}_t^{same}$, let $R_{t-1}(\mathrm{X}_i) = [\ell_i, m_i]$ be the predicated range of firing outputs in sub-round $(t-1,2)$. Then the expected number of firing outputs in sub-round $(t,2)$ is in the range $[p \cdot \ell_i \ , \ p \cdot m_i]$. For every $\mathrm{X}_i \in \mathcal{X}_t^{same}$, let $R_t(\mathrm{X}_i) = [(1 - 1/\log^3 n) \cdot p\ell_i \ , \ (1 + 1/\log^3 n) \cdot pm_i]$.

We now observe that if the expected number of firing outputs is too small for even one of the inputs in $\mathcal{X}_t^{same}$, then it implies a lower bound of $\Omega(\log n)$ for $\mathcal{ET}(\mathrm{N})$. Essentially this is because if this is the case, with good probability, 0 outputs will fire in round t, and a reset configuration identical to Y^0 will occur. This will keep occurring, causing the network to have large runtime.

▶ **Observation 26.** *For every $t \geq 1$, if there exists $\mathrm{X} \in \mathcal{X}_t^{same}$, such that the smallest value of $R_t(\mathrm{X}_i)$ is less then $1/\log^4 n$, then $\mathcal{ET}(\mathrm{N}) = \Omega(\log n)$.*

Proof. Let $\mathrm{X} \in \mathcal{X}_t^{same}$ be such that $R_t(\mathrm{X})$ is less then $1/\log^4 n$. Then, given that the inhibitors fire according to the prediction F_{t-1}^* in sub-round $(t-1,3)$, by Markov inequality, the probability that the number of firing outputs in sub-round $(t,2)$ is at least 1 is less then $1/\log^4 n$. In other words, the conditional probability (where we condition on the prediction for round $t-1$) that a reset where 0 outputs fire happens in sub-round $(t,2)$ is at least $1 - 1/\log^4 n$. However, by our inductive assumption $\widehat{F}(\mathrm{X}) = F_{t-1}^*$ must be correct with probability at least $1 - 1/\log n$. Hence, with probability at least $1 - \Theta(\log n)$ $Y^t = \vec{0}$ and a reset round occurs. With constant probability this occurs $\Omega(\log n)$ times before WTA is ever reached. The observation follows. ◀

Hence, from now on, we assume the complementary case that the number of predicted firing outputs in sub-round $(t, 2)$ is at least $1/\log^4 n$ for every $\mathrm{X} \in \mathcal{X}_t^{same}$. This allows us to show:

▶ **Claim 27.** *For every* $\mathrm{X} \in \mathcal{X}_t^{large}$

(1) *Given that the inhibitors fire according to* F_{t-1}^* *in sub-round* $(t-1, 3)$*, then with probability* $1 - 1/n$*, the number of firing outputs in sub-round* $(t, 2)$ *is in the range* $R_t(\mathrm{X})$*.*

(2) *The set of ranges* $R_t(\mathrm{X})$ *for* $\mathrm{X} \in \mathcal{X}_t^{large}$ *are all* a*-separated for some constant* a*.*

(3) $R_t(\mathrm{X})$ *has value at least* $\Omega(\log^7 n)$ *for every* $\mathrm{X} \in \mathcal{X}_t^{large}$*.*

Proof. Since for any $\mathrm{X} \in \mathcal{X}_t^{same}$ the predicted number of firing outputs is $\Omega(1/\log^4 n)$, and since the ranges are constant separated by our inductive assumption that the ranges $R_{t-1}(\mathrm{X})$ for $\mathrm{X} \in \mathcal{X}_{t-1}$ are separated, by omitting $\Theta(\log\log n)$ inputs from $\mathcal{X}_t^{same}$, the minimum number of firing outputs in the predicted ranges for the remaining set of inputs, namely, $\mathcal{X}_t^{large}$ is $\Omega(\log^7 n)$. Hence the true number of firing outputs is well concentrated around this expectation and so we have (1) by a Chernoff bound (Theorem 12).

Further, since we increase the width of the predicted range $R_t(\mathrm{X})$ by factor of at most $(1 + 1/\log^3 n)$ compared to the range $R_{t-1}(\mathrm{X})$, over all $O(\log\log n)$ rounds of prediction, the range is increased by at most a factor of $(1 + 1/\log^3 n)^{O(\log\log n)} \leq 1 + O(1/\log^2 n)$. Since the ranges have separation 2 in the initialization round, they remain constant separated in round t, giving (2). ◀

Predicting $\widehat{F}_t(\mathrm{X})$ given the predicted range $R_t(\mathrm{X})$

We first define the final subset $\mathcal{X}_t \subseteq \mathcal{X}_t^{large}$ of inputs for which round t is fully predicted (i.e., both the number of firing outputs in sub-round $(t, 2)$ and the states of the inhibitors in sub-round $(t, 3)$). The set $\mathcal{X}_t$ contains any $\mathrm{X} \in \mathcal{X}_t^{large}$ unless $R_t(\mathrm{X})$ intersects the critical range $K(z)$ for some convergence inhibitor $z \in C$. By Lemma 10, the firing state of each inhibitor $z \in C$ can be predicted with good probability as long as the number of firing outputs in previous sub-round is not in the critical range $K(z) = [k(z)/2, 2k(z)]$. In particular, if the range $R_t(\mathrm{X})$ falls below $k(z)/2$, then we predict that z does not fire in sub-round $(t, 3)$. On the other hand, if the range $R_t(\mathrm{X})$ falls above $2k(z)$, then we predict that z fires in sub-round $(t, 3)$. Regardless of the exact number of firing outputs in sub-round $(t, 2)$, since $R_t(\mathrm{X})$ does not intersect the critical ranges of the inhibitors of C, we can predict with good probability the firing states of C in sub-round $(t, 3)$ by Lemma 10. By Lemma 8, with probability at least $1 - 1/n$, all the stability inhibitors S fire in sub-round $(t, 3)$ and by Lemma 9, with good probability, no inhibitor in R fires. So overall we can predict all inhibitor behavior with good probability. With the above in place we are finally have that our inductive assumptions hold in round t. We summarize:

▶ **Lemma 28.** *For every* $t \geq 1$ *it holds that:*

(Q1) *For every* $\mathrm{X} \in \mathcal{X}_t$*, the predicted range of firing outputs* $R_t(\mathrm{X})$ *satisfies:*

$$\Pr[\widehat{R}_t(\mathrm{X}) \in R_t(\mathrm{X}) \;\mid\; \widehat{F}_{t-1}(\mathrm{X}) = F_{t-1}(\mathrm{X})] \geq 1 - 1/n \;. \tag{5}$$

(Q2) *The collection of predicted ranges* $R_t(\mathrm{X})$ *for* $\mathrm{X} \in \mathcal{X}_t$ *are all* a*-separated for some constant* a *and all have value at least* $\Omega(\log^7 n)$*.*

(Q3) *For every* $\mathrm{X} \in \mathcal{X}_t$*, the predicted firing pattern for the inhibitors satisfies*

$$\Pr[\widehat{F}_t(\mathrm{X}) = F_t(\mathrm{X}) \;\mid\; \widehat{R}_t(\mathrm{X}) \in R_t(\mathrm{X})] \geq 1 - 1/\log^3 n \;. \tag{6}$$

The final step before giving our expected time lower bound is to show that $\mathcal{X}_t$ is reasonably large, so we are able to keep predicting the behavior of N for a number of outputs round after round. This follows from a few simple observations:

▶ **Observation 29.** $|\mathcal{X}_t^{same}| \geq |\mathcal{X}_{t-1}|/\alpha$.

Recall that $\mathcal{X}_t^{same}$ consists of the largest subset of $\mathcal{X}_{t-1}$ with the *same* predicted inhibitor behavior F_{t-1}^* in round $t-1$. Naively, there are 2^α possible predictions for F_{t-1}^* which gives that $|\mathcal{X}_t^{same}| \geq |\mathcal{X}_{t-1}|/2^\alpha$. In order to obtain the much stronger bound above, we again use Lemma 10 which shows that, as long as $\widehat{R}_{t-1}(\mathrm{X})$ does not intersect the critical region of any $z \in C$, the inhibitors behave with good probability as linear threshold circuits and so there are only α possible predictions $F_{t-1}(\mathrm{X})$.

Proof. Since by Lemma 10 each inhibitor $z \in C$ behaves with probability $1 - \log^c n$ as a threshold network in sub-round $(t, 3)$ (so long that the number of firing outputs in sub-round $(t, 2)$ is not in the critical range $K(z)$), the total number of different inhibitor firing state configurations (different $F_{t-1}(\mathrm{X})$ vectors predicted in the previous step) is bounded by $|C|$. To see this, since conditioning on the prediction $R_t(\mathrm{X})$ being correct, there is at least one firing output in sub-round $(t-1, 2)$, the inhibitors of S will fire w.h.p. Further the inhibitors R never fire with good probability, so the only varying part in $F_{j-1}(\mathrm{X})$ is the prediction for C and as discussed there are only $|C| \leq \alpha$ such possible predictions. ◀

▶ **Observation 30.** $|\mathcal{X}_t^{large}| \geq |\mathcal{X}_t^{same}| - O(\log\log n)$.

This is immediate as $\mathcal{X}_t^{large}$ was derived by removing $\Theta(\log\log n)$ of the inputs with the smallest predicted range values from $\mathcal{X}_t^{same}$.

▶ **Observation 31.** $|\mathcal{X}_t| \geq |\mathcal{X}_t^{large}| - O(\alpha)$.

This follows as $\mathcal{X}_t$ is derived by removing all inputs from $\mathcal{X}_t^{large}$ where $R_t(\mathrm{X})$ overlaps the critical region of some $z \in C$. By (Q2) the $R_t(\mathrm{X})$ are all constant separated so there can be at most $|C| = O(\alpha)$ which overlap critical regions. We are now ready to show:

▶ **Lemma 32.** $\mathcal{ET}(\mathrm{N}) = \Omega(\log\log n/\log\alpha)$.

Proof. We can continue predicting the behavior of N up to round t until we have $|\mathcal{X}_t| = \Theta(\log\log n)$ (at which point $\mathcal{X}_t^{large}$ may be empty and so we will have to stop simulation). Further, as long as we can predict for t rounds, by Lemma 28 we will know with good probability that at least $\Omega(\log^7 n)$ outputs are still firing for all $\mathrm{X} \in \mathcal{X}_t$. So with good probability WTA is not reached for those inputs, giving a lower bound of $\Omega(t)$ rounds in expectation to solve WTA.

Set $t = c_1 \log\log n/\log\alpha$ for small enough constant c_1 and recall that we can assume $\alpha = O(\log^{c_2} n)$ for small constant c_2 since otherwise our runtime bound is $\Omega(1)$ and so holds vacuously. By Observations 29, 30, and 31 after t rounds we have:

$$\begin{aligned}|\mathcal{X}_t| &\geq \frac{|\mathcal{X}_0|}{\alpha^t} - t\cdot\alpha - t\cdot O(\log\log n)\\ &\geq \frac{\log n}{\log^{c_1} n} - \log\log n\cdot\log^{c_2} n - (\log\log n)^2 = \Omega(\log^{1-c_1} n)\end{aligned}$$

and hence can predict for at least t rounds. This completes the proof. ◀

Monotonicity property of basic WTA networks.

We show that the WTA dynamic is monotone so long as there is at least one firing output. Intuitively, we show that all basic WTA networks pick a single winner by monotonically decreasing the number of firing outputs until just a single output is firing. The number of firing outputs only ever increases if the network 'overshoots' the WTA state and has a round in which no outputs fire.

▶ **Lemma 33.** *For any basic WTA network* N*, as long as the number of firing outputs is more than one, their number is monotone non-increasing. In particular, if at least one output fires in round* t*, w.h.p. , an output that did not fire in that round, will not fire again in round* $t+1$*.*

Proof. Given input X with at least one firing input neuron, the network N must eventually converge so that in every round exactly 1 output fires w.h.p. Consider a round t in this *steady state period*. Since all outputs have the same parameters (e.g., edge weights and bias values) and since the weight of the self-loop is positive, if output y_i fires in round t, it is at least as likely to fire in round $t+1$ as output y_j for any $j \neq i$. Additionally, conditioned on the configuration of the inhibitors in time t, the probability that each output fires in round $t+1$ is independent. Hence, it must be that w.h.p., if y_i fired in round t, it continues to fire in round $t+1$ and each y_j, which did not fire in round t does not fire in round $t+1$ with high probability.

Further, consider any round t with at least one firing output. Since all connections from the output layer are excitatory, the probability that any inhibitor in Z fires at the end of round t is at least as large as it is in the steady state of the network, and hence any output that does not fire in round t does not fire in round $t+1$ w.h.p. ◀

B.6 Complete Proof for High Probability Lower Bound (Lemma 11)

Let $Q_Y \subseteq \{0,1\}^n, Q_Z \subseteq \{0,1\}^\alpha$ be the vectors describing the firing states of the outputs and inhibitors in a given round. Let $Q = Q_Y \circ Q_Z \subseteq \{0,1\}^{n+\alpha}$ be a vector describing the firing states of the inhibitors and outputs. Let $P_{1,j}(Q)$ be the probability to achieve the WTA state in round j given Q, that is the probability that exactly one output fires in sub-round $(j,2)$ given that the firing states of the outputs (resp., inhibitors) in sub-round $(j-1,2)$ (resp., $(j-1,3)$) is Q_Y (resp., Q_Z). Similarly, let $P_{0,j}(Q)$ be the probability that *no output* fires in sub-round $(j,2)$ given Q, that is the probability that a reset event happens. Finally, let $P_{01,j}(Q)$ be the probability that a reset event or a WTA event happens in round j given that configuration in round $j-1$ is Q, hence $P_{01,j}(Q) = P_{1,j}(Q) + P_{0,j}(Q)$. We begin by claiming the following.

▶ **Claim 34.** *For every round* j *and for every vector* $Q \in \{0,1\}^{n+\alpha}$ *in which there are at least two firing outputs (i.e.,* Q *is neither a WTA state nor a reset state), and such that* $P_{01,j}(Q) \geq \Theta(1/\log\log n)$*, it holds that* $P_{0,j}(Q) \geq \Theta(1/(\log\log n)^3)$*.*

Proof. Since $P_{01,j}(Q) = P_{0,j}(Q) + P_{1,j}(Q)$, if $P_{0,j}(Q) \geq P_{01,j}(Q)/2$, then we are done. Hence, we can assume from now on that $P_{1,j}(Q) = \Theta(1/\log\log n)$. We will show that $P_{0,j}(Q) \geq P_{1,j}(Q)/(\log\log n)^2$, which will establish our claim.

Let p be the firing probability of an active output[9] in sub-round $(j,2)$ given Q and let $k \geq 2$ be the number of outputs that fire in round $j-1$ as specified by Q. Since Q has at

[9] Recall that an output is active in round j if it fires in sub-round $(j-1,2)$.

least two firing outputs, w.h.p., only active outputs (those that fire in the previous round) can fire in the next round. The probability that the WTA state is achieved in round j is $P_{1,j}(Q) = k \cdot p \cdot (1-p)^{k-1}$ and the probability that a reset is achieved in round j is $P_{0,j}(Q) = (1-p)^k$.

We consider two cases depending whether the firing probability p is large or small. First, assume that $p \geq 0.1$ and set $r = c/\log\log n$. Since $P_{1,j}(Q) \geq r$, we have that $1 - p \geq r/k$. We also have:

$$k(9/10)^{k-1} \geq k \cdot p \cdot (1-p)^{k-1} \geq r,$$

and hence $k \leq \Theta(\log\log n)$. Overall, $P_{0,j}(Q)/P_{1,j}(Q) = (1-p)/(kp) \geq (1-p)/k \geq r/k^2 \geq c/(\log\log n)^2$. Next, consider the complementary case where $p < 0.1$. Letting $y = kp/2$,

$$y \cdot e^{-y} \geq (kp/2)(1-p)^{k/2} \geq (k/2)p(1-p)^{k-1} \geq r/2,$$

hence $y \leq 2\log 1/r = \Theta(\log\log\log n)$. Overall,

$$P_{0,j}(Q)/P_{1,j}(Q) = (1-p)/kp \geq \Theta(1/\log\log\log n).$$ ◀

The Execution Tree

A key tool used in this section is the notion *execution tree* that captures all possible transcripts that can evolve in a window of DH rounds when starting with the initial configuration C_0. The execution tree T is a tree of depth DH where each layer j corresponds to round j when running the network on the initial configuration C_0. Each node in T is labeled by an $(n+\alpha)$-length binary vector describing the firing configurations (or states) of the outputs and the inhibitors in a given round, and the edges are labelled by the transition probabilities. Hence, this tree describes all the possible firing states in a span of DH rounds when starting from the initial configuration C_0 (for which the time it takes to achieve WTA with constant probability is at least DC). The root r is labeled by the zero vector (since in round 0, no output fires and hence w.h.p also no inhibitor fires). For every $j \geq 2$, every node u in layer j is labeled by a vector $Q(u) = Q_Y(u) + Q_Z(u) \in \{0,1\}^{n+\alpha}$ describing the firing status of the outputs and the inhibitors in round j. Hence, each node has $2^{n+\alpha}$ children in the configuration tree. Every edge $e = (\pi(u), u)$ connecting u to its parent $\pi(u)$ in T is labeled by a probability $p(e)$ that the firing configuration in round j is $Q(u)$ given that the configuration in round $j-1$ is $Q(\pi(u))$.

Let $T_d(u)$ be the subtree of depth d rooted at u. When d is omitted $T(u)$ is simply the entire subtree of u in T.

For a leaf node $\ell \in T$, let $\mathcal{P}(\ell) = [r = u_0, u_1, \ldots, u_{DH}]$ be the path connecting ℓ to the root r in T. Let $p_{leaf}(u)$ be the probability that starting from r the firing configuration in each round $j \in \{0, \ldots, DH\}$ is $Q(u_j)$. Since there is an independence between the coin flips in every round j given the configuration in round $j-1$, we get that

$$\begin{aligned}
&p_{leaf}(u) = \\
&\prod_{j=0}^{DH} \Pr[Q_Y(u_j) \text{ in round } (j,2) \mid Q_Y(u_{j-1}), Q_Z(u_{j-1}) \text{ in rounds } (j-1,2),(j-1,3)] \\
&\cdot \Pr[Q_Z(u_j) \text{ in sub-round } (j,3) \mid Q_Y(u_j) \text{ in sub-round } (j,2)] \\
&= \prod_{j=1}^{DH} p(e_j) \quad \text{where} \quad e_j = (u_j, u_{j+1}).
\end{aligned}$$

For a node $u \in T$, let $Leaf(u)$ be the set of leaves in $T(u)$ and define

$$p_{node}(u) = \sum_{\ell \in Leaf(u)} p_{leaf}(u) \ ,$$

and for a subset of nodes U, let $p_{node}(U) = \sum_{u \in U} p_{node}(u)$. It is convenient to view $p_{node}(u)$ as the *weight* of tree $T(u)$. Hence, the weight of T is 1. In the same spirit, for a given subset of nodes U_i whose subtrees in T are vertex disjoint, we view $\sum_{u \in U_i} p_{node}(u)$ as the weight of the forest $\bigcup_{u \in U_i} T(u)$. We would like to show that:

$$\sum_{u \in Leaf(r)} \{p_{leaf}(u) \mid u \text{ is a WTA node } \} < 1 - 1/n^c \ , \tag{7}$$

In the next paragraphs, we will find a collection of non-WTA leaf nodes of large weight, i.e. of weight at least $1/n^2$ which will establish Eq. (7) for $c > 2$. To do that, we iteratively traverse the tree T from root to leaves, omitting undesired subtrees (and hence also leaf nodes) through the journey. This traversal is done in an asynchronous manner in the following sense: there are times that for a given node u in layer j, we move to a subset of its children in layer $j+1$, we call this move a *small jump* in the tree. In contrast, there are cases in which from a given node u in layer t, we jump DC layers in the subtree $T(u)$ and proceed the traversal from a subset of leaf nodes in the tree $T_{DC}(u)$ of depth DC, we call such a move a *large jump*. In the analysis part we will claim that by eliminating nodes in the tree T, we do not loose much weight, to deal with the fact that there are two types of jumps: small and large, we will employ an amortization claim that will enable us to bound the loss of weight layer by layer. See Fig. 3, for an illustration of the Execution Tree.

In each iteration $j \in \{1, \ldots, DH\}$, we maintain a collection of *non-WTA* nodes U_j whose subtrees in T are vertex disjoint. The final set U_{DH} will be a set of non-WTA leaf nodes for which we will show that their weight is at least $1/n^2$. Starting with $U_0 = \{r\}$, in every iteration $j \in \{1, \ldots, DH\}$, we have a set of nodes U_j that satisfy the following:

(A1) The subtrees $T(u)$, $u \in U_j$, are vertex-disjoint.
(A2) The distance of each node $u \in U_j$ from r is at least j.
(A3) No node in U_j is a WTA node.

In the high level, the nodes U_{j+1} are the leaf nodes of subtrees rooted at the nodes $u \in U_j$. Particularly, from each node $u \in U_j$, when constructing U_{j+1}, we omit part of the subtree $T(u) \subseteq T$ and replace u by a subset of nodes $V(u)$ in the subtree of u in T. The nodes $V(u)$ are subset of the leaf nodes of the subtree $T_{d(u)}(u)$ of depth $d(u)$ rooted at u. The value of the depth $d(u)$ is set to be either 1 or DC [10] depending on the configuration stored at node u. That is, either the nodes $V(u)$ are a subset of the children of u or that they are subset of the leaf nodes of the DC-depth tree rooted at u.

In the first case where $d(u) = 1$, we will show that we loose only $\Theta(1/\log\log n)$ of the weight of the tree $T(u)$, hence we keep $1 - \Theta(1/\log\log n)$ fraction of the weight. In the second case, we will show that we keep $\Theta(1/\log\log n)$ fraction of the weight of $T(u)$. The key observation here is to note that this cannot happen more than DH/DC times in a given branch, since the depth of the sub-tree of u is DC. In other words, on average, we maintain $\Theta(1/\log\log n)^{1/DC}$ of the weight per layer of the subtree $T_{DC}(u)$, and hence overall, after DH iterations, we maintain $1/n^2$ fraction of the total weight.

We first eliminate from the tree T all nodes u such that $Q_Y(u) = \vec{0}$ but $Q_Z(u) \neq \vec{0}$. Since the bias value of the inhibitors in $\Omega(\log n)$, we know that if no output fires in round j, then

[10] To be more precise it is either 1 or $\min\{DC, DH - dist(r, u, T)\}$.

w.h.p. no inhibitor fires in that round. Let T' be the resulting tree. We first observe that by that step, we eliminate only $1/n^c$ of the total weight of the tree T.

▶ **Observation 35.** *The total weight of r in T' is at least $1 - 1/n^c$.*

From now on, we consider the tree T' and describe the iterative construction of the set U_j in details. Let $U_0 = \{r\}$. For $j \geq 1$ given U_j, the set U_{j+1} is obtained by defining for each node $u \in U_j$, a subset of non-WTA nodes $V(u)$ as described next.

Case 1: u is a reset node. Set the depth of the subtree to be $d(u) = \min\{DH - dist(u, r, T), DC\}$ and let $V(u)$ be the non-WTA nodes in the leaf nodes of $T_{d(u)}(u)$.

Since u is not a WTA node, it remains to consider the case where the number of active outputs in $Q(u)$ is at least 2. Recall that $P_{01,j}(Q(u))$ be the probability of achieving WTA or reset in round $t+1$ given the configuration in round t is $Q(u)$. We distinguish between two cases depending on the value of $P_{01,j}(Q(u))$.

Case 2.1: $P_{01,j}(Q(u)) \geq \Theta(1/\log\log n)$. Let $V'(u)$ be the children of u in T that are reset-nodes. For each reset-node $w \in V'(u)$, let $V(w)$ be the non-WTA nodes in the leaf nodes of $T_{d(u)-1}(w)$ and let $V(u) = \bigcup V(w)$.

Case 2.2: $P_{01,j}(Q(u)) < \Theta(1/\log\log n)$. Let $V(u)$ be the children of u that have at least 2 active outputs in $Q(v)$ (hence $d(u) = 1$). This completes the definition of U_{j+1}.

To bound the weight of U_{DH}, we make use of the following claims that show that we do not loose too much weight in this traversal. Consider a node u and let $NW_{DC}(u)$ be the set of non-WTA leaves of the tree $T_{DC}(u)$.

▶ **Claim 36.** *If u is a reset node, then $p_{node}(NW_{DC}(u)) \geq c' \cdot p_{node}(u)$, for some constant c'.*

Proof. Let j be the layer of node u. Then by the selection of the initial configuration C_0, we know that the time it takes to achieve WTA with constant probability c when starting from C_0 is strictly larger than DC. Since a reset node is labelled with this same initial configuration, we get that $p_{node}(NW_{DC}(u)) \geq c' \cdot p_{node}(u)$ for $c' = 1 - c$. ◀

▶ **Claim 37.** *Let u be a node in layer j that satisfies Case (1) or Case (2.1), then $p_{node}(V(u)) \geq \Theta((1/\log\log n)^3) \cdot p_{node}(u)$.*

Proof. If u satisfies Case (1), the claim follows immediately by Cl. 36. We now consider the case where u satisfies Case (2.1). Recall that in this case the number of active outputs in $Q(u)$ is at least 2. Let A_0, A_1 be the set of children of u that are reset nodes, WTA nodes respectively. Let $A_{0,1} = A_0 \cup A_1$.

Then, since u satisfies Case (2.1), $p_{node}(A_{0,1}) \geq \Theta(1/\log\log n) \cdot p_{node}(u)$. In addition, since in $Q(u)$ there are at least two firing outputs, we can safely apply Cl. 34, to have that $p_{node}(A_0) \geq \Theta(1/(\log\log n)^2) \cdot p_{node}(A_1)$. Combining these two inequalities, we get that

$$p_{node}(A_0) \geq \Theta(1/(\log\log n)^3) \cdot p_{node}(u).$$

Next, by using Cl. 36, for every node $v \in A_0$ (which is a reset node), we have that $p_{node}(NW_{DC-1}(v)) \geq c' \cdot p_{node}(v)$. All together, we get that

$$\begin{aligned} p_{node}(NW_{DC}(u)) &\geq \sum_{v \in A_0} p_{node}(NW_{DC-1}(v)) \\ &\geq c' \cdot p_{node}(A_0) \geq \Theta(1/(\log\log n)^3) \cdot p_{node}(u) \ . \end{aligned}$$

Since $V(u) = A_0$, the claim follows. ◀

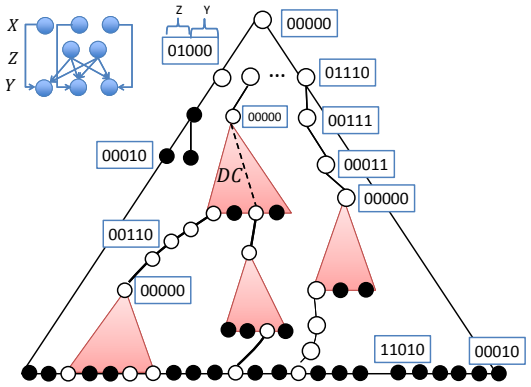


Figure 3 The Execution Tree. Shown is a schematic illustration of the Execution Tree for a small network with two inhibitors and three outputs. Every node u in layer j is labeled by a vector of length 5 describing an optional firing state for the inhibitors and outputs in round j. Each node has 2^5 children – covering all possible firing behaviors in round $j+1$. The weight nodes are non WTA nodes and the black nodes are the WTA nodes. When arriving a reset node u, a large jump is made by considering the leaf nodes of $T(u)$. When arriving a non-WR node, a small jump is made by considering subset of its children.

▶ **Claim 38.** *Let u be a node that satisfies Case (2.2), then* $\sum_{w\in V(u)} p_{node}(w) \geq 1 - \Theta(1/\log\log n)\cdot p_{node}(u)$.

Proof. By the definition of u, $P_{01,j}(Q(u)) < \Theta(1/\log\log n)$. Hence, letting $V(u)$ be the children of u that have at least 2 active outputs in $Q(v)$ (hence $d(u)=1$), we have that $\sum_{w\in V(u)} \geq 1 - \Theta(1/\log\log n)\cdot p_{node}(u)$. ◀

Starting from a tree of weight 1, we would like to show that at the end of the process after at most DH iterations, the total weight of the leaf nodes U_{DH} is at least $1/n^2$. We now use Cl. 37 and 38 to prove the lower bound. By Cl. 37, when we consider $u \in U_j$ that satisfies either case (1) or case (2.1), we keep $\Theta(1/(\log\log n)^3)$ fraction of the weight but enjoy a large jump of DC layers in the sub-tree $T(u)$. Hence, on average, we keep $\Theta(1/(\log\log n)^3)^{1/DC}$ fraction of the weight of $T(u)$ per layer. By Cl. 38, in type (2), we keep at least $1-\Theta(1/\log\log n)$ fraction of the weight of $T(u)$ when moving from a node u in layer i to a subset of its children $V(u)$ in layer $i+1$. Hence, on average in every iteration, we keep at least

$$\max\{(c/(\log\log n)^3)^{1/DC}, 1 - c/\log\log n\}$$

fraction of the weight of the current forest. Hence, after $DH = DC\cdot\Theta(\log n/\log\log\log n)$ iterations, our total weight of the leaf set U_{DH} is at least

$$(\max\{(c/(\log\log n)^3)^{1/DC}, 1 - c/\log\log n\})^{DH} \geq 1/n^2.$$

C Extension to Excitatory Auxiliary Neurons

In this section, we consider the more general case where the auxiliary neurons can be either excitatory or inhibitory. Let α denote their number. We assume that outputs with no active input are not allowed to fire. Hence, in a given sub-round $(t,2)$, we consider two types of outputs that might fire: *active* outputs – those that fire in the previous round and hence

have a positive feedback via the self-loop; and *inactive* outputs – those that did not fire in the previous round. Whereas in the inhibitory case, we could show that the dynamic is monotonic – hence incative outputs do not fire with high probability, here it is not the case. Specifically, it might be the case that the level of inhibition during the process to achieve the WTA state is *lower* than that in steady-state and hence inactive outputs (outputs that did not fire in the previous rounds) join the game in later rounds. In our lower bound proofs, we heavily used the monotonicity property as it allowed us focus only on the active outputs (those that fired in the previous rounds) and totally neglect the inactive ones. In this section, we revise the claims that are based on the monotonicity lemma and adapt the proof to the general case of excitatory and inhibitory neurons.

C.1 Extensions for the Lower Bound for Expected Time

We classify the auxiliary neurons as before into three classes S, C and R. Note that all the proofs that concern the predictability of the inhibitors, i.e., Lemmas 8,9,10 depend only on the potential functions of the inhibitors and not on their effect on the outputs. Since the excitatory auxiliary neurons have exactly the same potential functions, the proofs follow immediately.

The main adaptation is in the second part where we use the predictability of the auxiliary neurons to predict the network for at least one input configuration. We proceed by bounding the gap in potentials between active outputs and inactive outputs by showing that the weight of the self-loop is large.

▶ **Observation 39.** $\mathrm{w}^{\mathrm{self}} \geq 2c \cdot \log n$.

Proof. In the steady state situation, there exists one leader u that fires in each round w.h.p. $1 - 1/n^c$ for polynomially many rounds. On the other hand, all other outputs v that do not have the positive feedback from the self-loop fire with probability $1/n^c$. Hence for such a round t in steady state, we have: $pot_t(u) \geq c \log n$ and $pot_t(v) \leq -c \log n$. We get that $\mathrm{w}^{\mathrm{self}} = pot_t(u) - pot(v) \geq 2c \log n$. The observation follows. ◀

An immediate corollary of that is the following:

▶ **Corollary 40.** *Consider a sub-round $(t, 2)$ and let F_{t-1} be the firing configuration of the auxiliary neurons in sub-round $(t-1, 3)$. If the firing probability of an inactive output v (output that did not fire in the previous sub-round $(t-1, 2)$) in sub-round $(t, 2)$ is at least $1/n^c$, then the firing probability of an active output u in sub-round $(t, 2)$ is $\geq 1 - 1/n^c$.*

Proof. Since all outputs have the same connections to the auxiliary neurons, only difference in the potential of an inactive output and an active output is the weight of the self-loop. Hence, $pot_t(u) = pot_t(v) + \mathrm{w}^{\mathrm{self}} \geq -c \log n + 2c \log n \geq c \log n$, where the first inequality follows by plugging Obs. 39 and using the fact that the firing probability of v is $1/(1+e^{-pot_t(v)}) \geq 1/n^c$. Thus, u fires with probability $1/(1 + e^{-c \log n}) = 1 - 1/n^c$. ◀

We now consider the second part of the lower bound where we predict $\Omega(\log \log n / \log \alpha)$ rounds of the network for at least one density input class. Since in the zero round no-output fires and w.h.p. also no auxiliary neuron is firing (since their bias value is $\omega(\log n)$), predicting the number of firing outputs in round 1 is exactly the same as in the only-inhibitor case.

Predicting the number of firing outputs in round $t \geq 2$

We first define a subset of inputs $\mathcal{X}_t^{large} \subseteq \mathcal{X}_{t-1}$ for which we can predict the behavior of the outputs in the network N in round t. Let $\mathcal{X}_t^{same} \subseteq \mathcal{X}_{t-1}$ be the largest subset of inputs whose predicted firing vector $F_{t-1}(\mathrm{X})$ for the auxiliary neurons in round $t-1$ is the *same*, and denote this common firing vector by F_{t-1}^*. Let $\mathcal{X}_t^{large}$ be the set of inputs in $\mathcal{X}_t^{same}$ after omitting $\Theta(\log\log n)$ inputs with the smallest range value in round $t-1$. Eventually we will show that $\mathcal{X}_t^{large}$ is a reasonably large set of inputs compared to $\mathcal{X}_{t-1}$, and hence we can continue predicting behavior for at least some inputs for a large number of rounds. But first we show how to predict $R_t(\mathrm{X})$ for every input $\mathrm{X} \in \mathcal{X}_t^{large}$.

Let p' be the firing probability that an inactive output (one with $y_j^{t-1} = 0$) fires in sub-round $(t,2)$ given that the inhibitors fired in sub-round $(t-1,3)$ according to F_{t-1}^*. Since all inputs in $\mathcal{X}_t^{same}$ have the same predicted firing vector F_{t-1}^*, in each of them, an inactive output fires in sub-round $(t,2)$ with probability p'. Let p be the corresponding firing probability of an active output. We now consider two cases depending on the value of p'. If $p' < 1/n^c$, we predict that no inactive output fires in that round. Note that this prediction holds with probability $\geq 1 - 1/n^{c-1}$. In such a case we only predict the range for the active outputs in the exact same manner as before. Note that when we predicted the range of firing active outputs in the previous section, we did not use the fact that the auxiliary neurons are inhibitory, only that all competing outputs whose cardinality is to be estimated fire with the same probability in that round.

Next, we consider the more interesting case where $p' \geq 1/n^c$, that is the inactive outputs have a fair chance of firing in sub-round $(t,2)$. Here, we make use of Lemma 40 that says that with probability at least $1 - 1/n^{c-1}$, all active outputs (i.e., that fired in round $t-1$) fire in sub-round $(t,2)$ as well. Let $k = 2^i$ be the number of active inputs in the vector X. Let $E_{t-1} = E(\widehat{R}_{t-1}(\mathrm{X}) \mid F_{t-2}(X))$ be the expected number of firing outputs in sub-round $(t-1,2)$ given the predicted firing vector $F_{t-2}(X)$. Then, the expected number of firing outputs in sub-round $(t,2)$ is

$$E(\widehat{R}_t(\mathrm{X}) \mid F_{t-1}(X)) = E_{t-1} + p' \cdot (k - E_{t-1}) = (1-p') \cdot E_{t-1} + p' \cdot k.$$

▶ **Claim 41.** *Let* $\mathrm{X}_1, \mathrm{X}_2 \in \mathcal{X}_t$ *be such that* $||\,\mathrm{X}_1\,||_1 \geq 2||\,\mathrm{X}_2\,||_1$. *Then*

$$E(\widehat{R}_t(\mathrm{X}_1) \mid F_{t-1}(\mathrm{X}_1)) \geq 2E(\widehat{R}_t(\mathrm{X}_2) \mid F_{t-1}(\mathrm{X}_2)).$$

Proof. We will prove by induction on the number of rounds t. Let $k_j = ||\,\mathrm{X}_j\,||_1$ and $E_{j,t} = E(\widehat{R}_t(\mathrm{X}_j) \mid F_{t-1}(\mathrm{X}_j))$ for $j \in \{1,2\}$.

Since $\mathrm{X}_1, \mathrm{X}_2 \in \mathcal{X}_t$, it holds that $\mathrm{X}_1, \mathrm{X}_2 \in \mathcal{X}_\ell$ for every $\ell \in \{1,\ldots,t\}$ hence $F_\ell(\mathrm{X}_1) = F_\ell(\mathrm{X}_2)$ for every $\ell \in \{1,\ldots,t\}$. For the base of the induction of round $t = 1$, this clearly holds since $E_{0,t} = p_0 \cdot k_j$, $j \in \{1,2\}$, where p_0 is the firing probability of an output where in the previous round no one fired. Assume the claim holds up to round $t-1$. We have that $E_{j,t} = E_{j,t-1} + p' \cdot (k_j - E_{j,t-1}) = (1-p')E_{j,t-1} + p'k_j$, for $j \in \{1,2\}$. By the induction assumption for $t-1$, we get $E_{1,t-1} \geq 2 \cdot E_{2,t-1}$ and by definition $k_1 \geq 2 \cdot k_2$, overall $E_{1,t} \geq 2E_{2,t}$ as required. ◀

We get that the expected number of firing outputs (conditioned on the predictions) are 2-separated. Now, we can claim exactly as before that all these expected values should be $\Omega(1/\log^4 n)$ as otherwise there is at least one input configuration for which there is a reset (i.e., in the next round no output fires) for $\Omega(\log n)$ times (see Obs. 26).

Since all expected predictions for the number of firing outputs are $\Omega(1/\log^4 n)$, by removing the $\Theta(\log\log n)$ inputs from $\mathcal{X}^{same}$ (i.e., as given by set $\mathcal{X}^{large}$), we get that all

expected numbers of firing outputs are $\Omega(\log^7 n)$ and hence the random variables $\widehat{R}_t(\mathrm{X})$ are well concentrated around their expectation. The remaining proof goes exactly the same as in the inhibitory-case.

C.2 Extensions for the Lower Bound for High Probability Time

We define the *weak* WTA state to be state in which exactly one active output is firing (but possibly many inactive firing outputs). Whenever we use the notion of WTA nodes in the proof of Lemma 11, we now use the notion of weak WTA nodes instead. The definition of a reset node remains as is, i.e., a node u such that in its configuration $Q(u)$ no output (of any type) fires.

Note that the lower bound proof for the expected time implies that there is an input X_0 such that with a good probability after $t = \Omega(\log\log n/\log\alpha)$ rounds there are still $\Omega(\log n)$ competing outputs. After $t+1$ rounds, either we can assume w.h.p. that no inactive output fires or that all the $\Omega(\log n)$ active outputs fire. Hence, the lower bound implies that after $t+1$ rounds, with good probability, the number of firing active outputs is $\Omega(\log n)$, implying that the network is in a *weak* WTA state. Let $P_{1,j}(Q)$ be the probability that exactly one *active* output fires in sub-round $(j,2)$ given that the auxiliary neurons fire in round $j-1$ according to Q. Similarly, let $P_{0,j}(Q)$ be the probability that *no active output* fires in sub-round $(j,2)$ given Q. Finally, let $P_{01,j}(Q)$ be the probability that at most one active output fires in round $(j,2)$ given that configuration in round $j-1$ is Q, hence, $P_{01,j}(Q) = P_{1,j}(Q) + P_{0,j}(Q)$. Since we consider only the active outputs, Cl. 34 follows as is. We now claim the following.

▶ **Corollary 42.** *For every round j and for every vector $Q \in \{0,1\}^{n+\alpha}$ in which there are at least two active firing outputs and such that $P_{01,j}(Q) \geq \Theta(1/\log\log n)$, it holds that there is a (total) reset in round j (i.e., no output fires) with probability at least $\Theta(1/(\log\log n)^3)$.*

Proof. Since in Q there are at least two firing *active* outputs, by Cl. 34, $P_{0,j}(Q) \geq \Theta(1/(\log\log n)^3)$. Hence the probability that no active output fires is at least $\Theta(1/(\log\log n)^3)$. We now claim that the probability that also no inactive output fires is at least $1-1/n^{c-1}$. Hence, by the independence between the output decisions (given the firing states of the inhibitors), we get that the probability that no output fires is at least $\Theta(1/(\log\log n)^3)$ as required.

Assume towards contradiction that inactive output fires with probability $\geq 1/n^c$. By Cor. 40, we get that an active output fires with probability at least $1-1/n^c$. Since in the previous round there are at least two firing *active* outputs, we get that with probability $\geq 1-1/n^c$ there are at least two firing outputs in sub-round $(j,2)$, contradiction to the assumption that $P_{01,j}(Q) \geq \Theta(1/\log\log n)$.

Thus we get that each inactive output fires with probability $< 1/n^c$, and with probability $\geq 1-1/n^c$ no inactive output fires. The claim follows. ◀

Equipped with Cor. 42 and the lower bound for expected time, we can now use the execution tree to show that the weight of non weak-WTA nodes is at least $1/n^2$. The same idea generally holds up to few adaptations. Recall that in our execution tree traversal, at step j we obtain a collection of non weak WTA nodes. That is nodes u with configuration $Q(u)$ which either there are at least two active outputs that are firing. For $j \geq 1$ given U_j, the set U_{j+1} is obtained by defining for each node $u \in U_j$, a subset of non weak WTA nodes $V(u)$ as described next.

Case 1: u is a reset node. Set the subtree depth $d(u) = \min\{DH - dist(u, r, T), DC\}$ and let $V(u)$ be the non weak WTA nodes in the leaf nodes of $T_{d(u)}(u)$. By the lower bound proof, the set $V(u)$ captures $1 - 1/\log n$ of the probability mass in $T(u)$.

Since u is a non weak WTA node, it remains to consider the case where the number of active firing outputs in $Q(u)$ is at least 2. Recall that $P_{0,1}(Q(u))$ is the probability that in sub-round $(j, 2)$ at most one active output fires given that the configuration in round $j-1$ is $Q(u)$.

Case 2.1: $P_{0,1}(Q(u)) \geq \Theta(1/\log\log n)$. Let $V'(u)$ be the children of u in T that are reset-nodes. For each reset-node $w \in V'(u)$, let $V(w)$ be the non-WTA nodes in the leaf nodes of $T_{d(u)-1}(w)$ and let $V(u) = \bigcup V(w)$.
By Cl. 42, since the number of firing active outputs in $Q(u)$ is at least 2 and since $P_{0,1}(Q(u)) \geq \Theta(1/\log\log n)$, the probability for a (total) reset in the next round is at least $\Theta(1/(\log\log n)^3)$ and hence $V'(u)$ captures $\Theta(1/(\log\log n)^3)$ of the probability mass in $T(u)$. This will allow us to follow the same argument as before when following the case 2.1.

Case 2.2: $P_{0,1}(Q(u)) < \Theta(1/\log\log n)$. Let $V(u)$ be the children of u that have at least 2 *active* outputs in $Q(v)$ (hence $d(u) = 1$). Since $P_{0,1}(u) \leq \Theta(1/\log\log n)$, we capture $1 - \Theta(1/\log\log n)$ of the weight of the tree $T(u)$. This completes the definition of U_{j+1}. The argument that uses this case follows now the exact same line. In sum, either we capture only $\Theta(1/\log\log n)$ of the probability mass in such a case we have a large jump in the tree or that we capture $1 - \Theta(1/\log\log n)$ of the probability mass. As before using the amortization argument, overall the number of non weak WTA nodes can be bounded by $\geq 1/n^2$. The completes the extension to excitatory auxiliary neurons.

Mutation, Sexual Reproduction and Survival in Dynamic Environments

Ruta Mehta[1], Ioannis Panageas[2], Georgios Piliouras[3], Prasad Tetali[4], and Vijay V. Vazirani[5]

1 **University of Illinois Urbana-Champaign, USA**
rutamehta@cs.illinois.edu
2 **MIT, Cambridge, USA and**
Singapore University of Technology and Design, Singapore
panageasj@gmail.com
3 **Singapore University of Technology and Design, Singapore**
georgios.piliouras@gmail.com
4 **Georgia Institute of Technology, Atlanta USA**
tetali@math.gatech.edu
5 **Georgia Institute of Technology, Atlanta USA**
vazirani@cc.gatech.edu

Abstract

A new approach to understanding evolution [34], namely viewing it through the lens of computation, has already started yielding new insights, e.g., natural selection under sexual reproduction can be interpreted as the *Multiplicative Weight Update* (MWU) Algorithm in coordination games played among genes [8]. Using this machinery, we study the role of mutation in changing environments in the presence of sexual reproduction. Following [35], we model changing environments via a Markov chain, with the states representing environments, each with its own fitness matrix. In this setting, we show that in the absence of mutation, the population goes extinct, but in the presence of mutation, the population survives with positive probability.

On the way to proving the above theorem, we need to establish some facts about dynamics in games. We provide the first, to our knowledge, polynomial convergence bound for noisy MWU in a coordination game. Finally, we also show that in static environments, sexual evolution with mutation converges, for any level of mutation.

1998 ACM Subject Classification G.1.5 [Numerical Analysis]: Roots of Nonlinear Equations–Convergence, Systems of equations; J.3 [Computer Applications]: Life and medical sciences–Biology and genetics

Keywords and phrases Evolution, Non-linear dynamics, Mutation

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.16

1 Introduction

Evolution has been the subject of intensive scientific investigation since the early 19th century, yet many of its critical elements still remain under debate. A new, potent approach to studying evolution was initiated by Valiant [34], namely viewing it through the lens of computation. This viewpoint has already started yielding concrete insights by translating qualitative hypotheses in biological systems to provable computational properties of Markov chains and other dynamical systems [18, 8, 22, 24, 30, 31], which are standard hallmarks of TCS research. We build on this direction whilst focusing on the challenge of evolving environments.


© Ruta Mehta, Ioannis Panageas, Georgios Piliouras, Prasad Tetali, and Vijay V. Vazirani;
licensed under Creative Commons License CC-BY

Recent work due to Chastain, Livnat, Papadimitriou and Vazirani [8] linked natural selection under sexual reproduction (*sexual evolution*) to a tangential field, namely dynamics in games. Building on the work of Nagylaki [26], they showed that this process can be interpreted as the *Multiplicative Weight Update* (MWU) Algorithm [3], which we call discrete replicator dynamics, in coordination games played among genes. This connection opens up doors for applying tools from game theory and dynamical systems to understanding these fundamental processes. Another important question is the role of mutation, especially in the presence of changes to the environment. In the case of asexual reproduction, this was studied by Wolf, Vazirani, and Arkin [35]. They modeled a changing environment via a Markov chain and described a model in which in the absence of mutation, the population goes extinct, but in the presence of mutation, the population survives with positive probability.

In this paper we study the next natural question, namely the role of mutation in the presence of sexual reproduction. The Chastain et al. result shows that MWU (sex) tries to maximize fitness as well as entropy via the process of recombination in genes. The question arises whether this is enough to safeguard against extinction in a changing environment, or is mutation still needed.

As in Chastain et al. [8], we will consider a haploid organism with two genes. Each gene can be viewed as a player in a game and the alleles of each gene represent strategies of that player. Once an allele is decided for each gene, an individual is defined, and its fitness is the payoff to each of the players, i.e., both players have the same payoff matrix, and therefore it is a coordination/partnership game. Each state of the Markov chain represents an environment and has its own fitness matrix. We show the following under this model, where mutations are captured through a standard model appeared in [13]:

Informal Theorem 1. For a class of Markov chains (satisfying somewhat necessary conditions, see Section B), a haploid species under *sexual evolution*[1] without mutation dies out with probability one. In contrast, under sexual evolution with mutation the probability of long term survival is strictly positive.

For each gene, if we think of its allele frequencies in a given population as defining a mixed strategy, then after reproduction, the frequencies change as per MWU [8]. Furthermore, in the presence of mutation [13], every allele mutates to another allele of the corresponding gene in a small fraction of offsprings. As it turns out, in every generation the population size (of the species) changes by a multiplicative factor of the current expected payoff (mean fitness). Hence, in order to prove Theorem 1, we need to analyze MWU (and its variant which captures mutations) in a time-evolving coordination game whose matrix is changing as per a Markov chain.

The idea behind the first part of the theorem is as follows: It is known that MWU converges, in the limit, to a pure equilibrium in coordination games [22]. This implies that in a static environment, in the limit the population will be rendered monomorphic. Showing such a convergence in a stochastically changing environment is not straightforward. We first show that such an equilibrium can be reached fast enough in a static environment. We then appeal to the Borel-Cantelli theorem to argue that with probability one, the Markov chain will visit infinitely often and remain sufficiently long in one environment at some point and hence the population will eventually become monomorphic. An assumption in our theorem is that for each individual, there are bad environments, i.e., one in which it will go extinct.

[1] We refer to 'evolution by natural selection under sexual reproduction' by *sexual evolution* for brevity.

Eventually the monomorphic population will reach such an unfavorable environment and will die out.

Polynomial time convergence in static environment: For such a reasoning to be applicable we need a fast convergence result, which does not hold in the worst case, since by choosing initial conditions sufficiently close to the stable manifold of an unstable equilibrium, we are bound to spend super-polynomial time near such unstable states. To circumvent this we take a typical approach of introducing a small noise into the dynamics [32, 12, 14], and provide the first, to our knowledge, polynomial convergence bound for noisy MWU in coordination games; this result is of independent interest. We note that pure MWU captures frequency changes of alleles in case of infinite population, and the small noise can also be thought of as sampling error due to finiteness of the population. In the following theorem, dependence on all identified system parameters is necessary (see discussion in Section B).

Informal Theorem 2. In static environments under small random noise ($||.||_\infty = \delta$), sexual evolution (without mutation) converges with probability $1-\epsilon$ to a monomorphic fixed point in time $O\left(\frac{n \log \frac{n}{\epsilon}}{\gamma^4 \delta^6}\right)$, where n is the number of alleles, and γ the minimum fitness difference between two genotypes.

Although mutations seem to hurt mean population fitness in the short run in static environments, they are critical for survival in dynamic environments, as shown in the second part of Theorem 1; it is proved as follows. The random exploration done by mutations and aided by the selection process, which rapidly boosts the frequency of alleles with good mean fitness, helps the population survive. Essentially we couple the random variable capturing population size with a biased random walk, with a slight bias towards increase. The result then follows using a well-known lemma on biased random walks.

Robustness to mutations: Finally we show that the convergence of MWU (without mutation) in static environments [21, 19, 22] can be extended to the case where mutations are also present. The former result critically hinges on the fact that mean fitness strictly increases under MWU in coordination games, and thereby acts as a potential function. This is no more the case. However, using an inequality due to Baum and Eagon [4] we manage to obtain a new potential function which is the product of mean fitness and a term capturing diversity of the allele distribution. The latter term is essentially the product of allele frequencies.

Informal Theorem 3. In static environments, sexual evolution with mutation converges, for any level of mutation. Specifically, if we are not at equilibrium, at the next time generation at least one of mean population fitness or product of allele frequencies will increase.

Besides adding computational insights to biologically inspired themes, which to some extent may never be fully settled, we believe that our work is of interest even from a purely computational perspective. The nonlinear dynamical systems arising from these models are gradient-like systems of non-convex optimization problems. Their importance and the need to develop a theoretical understanding beyond worst case analysis has been pinpointed as a key challenge for numerous computational disciplines, e.g., from [2]:

> *"Many procedures in statistics, machine learning and nature at large – Bayesian inference, deep learning, protein folding – successfully solve non-convex problems ... Can we develop a theory to resolve this mismatch between reality and the predictions of worst-case analysis?"*

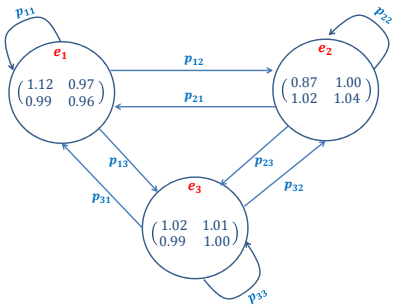


Figure 1 An example of a Markov Chain model of fitness landscape evolution.

Our theorems and techniques share this flavor. Theorem 1 expresses time-average efficiency guarantees for gradient-like heuristics in the case of time-evolving optimization problems, Theorem 2 argues about speedup effects by adding noise to escape out of saddle points, whereas Theorem 3 is a step towards arguing about robustness to implementation details. We make this methodological similarities more precise by pointing them out in more detail in the related work section (see discussion in Section 2).

Organization of the paper. The rest of the paper is organized as follows. In the next section we discuss the relevant literature. In Section 3 we provide formal description of the model we analyze. In Section 4, we provide an overview of the proofs of our main theorems. The proofs of main Theorems 1, 2, 3 and their formal statements appear in Section 6, 5 and 7 respectively. The omitted proofs can be found in the Appendix, along with explanation of the biological terms.

2 Related Work

In the last few years we have witnessed a rapid cascade of theoretical results on the intersection of computer science and evolution. Livnat et al. [17] introduced the notion of mixability, the ability of an allele to combine itself successfully with others. In [8] connections were established between sexual evolution and dynamics in coordination games. Meir and Parkes [24] has provided a more detailed examination of these connections. These dynamics are close variants of the standard (discrete) replicator dynamics [13]. Replicator dynamics is closely connected to the multiplicative weights update algorithm [14, 28]. In [22] Mehta et al. established that these systems converge for almost all initial conditions to monomorphic states. It is also possible to introduce connections between satisfiability and evolution [18] as well as understand the complexity of predicting the survival of diversity in complex species [23]. In [30] Panageas et al. shed light on the speed of asexual evolution (see also [31]). Wolf, Vazirani, and Arkin [35] analyzed models of mutation and survival of diversity also for asexual populations but the dynamical systems in this case are linear and the involved methodologies are rather different.

Introducing noise in non-linear dynamics have been shown to be able to simplify the analysis of nonlinear dynamical systems by "destroying" Turing-completeness of classes of dynamical systems and thus making the system's long-term behavior computationally

predictable [6]. Those techniques focus on establishing invariant measures for the systems of interest and computing their statistical characteristics. In our case, our unperturbed dynamical systems have exponentially many saddle points and numerous stable fixed points and species survival is critically dependent on the the amount of time that trajectories spend in the vicinity of these points thus much stronger topological characterizations are necessary. Adding noise to game theoretic dynamics [14, 1, 9] to speed up convergence to approximate equilibria in potential games is a commonly used approach in algorithmic game theory, however, the respective proof techniques and notions of approximation are typically sensitive to the underlying dynamic, the nature of noise added as well as the details of the class of games.

In the last year there has been a stream of work on understanding how gradient (and more generally gradient-like) systems escape out of the saddle fixed points fast [12, 15]. This is critically important for a number of computer science applications, including speeding up the training of deep learning networks. The approach pursued by these papers is similar to our own, including past papers in the line of TCS and biology/game theory literature [14, 28, 22]. For example, in [28, 22] it has been established that in non-convex optimization settings gradient-like systems (e.g., variants of Multiplicative Weights Updates Algorithm) converge for all but a zero measure of initial conditions to local minima of the fitness landscape (instead of saddle points even in the presence of exponentially many saddle points). Moreover, as shown in [14] noisy dynamics diverge fast from the set of saddle points whose Jacobian has eigenvalues with large positive real parts. Similar techniques and arguments can be applied to argue generic convergence to local minima of numerous other dynamics (including noisy/deterministic versions of gradient dynamics). Finally, using techniques developed in [28], one can argue that gradient dynamics converge to local minima with probability one in non-convex optimization problems even in the presence of continuums of saddle points [29], answering an open question in [15]. We similarly hope that techniques developed here about fast and robust convergence can also be extended to other classes of gradient(-like) dynamics in non-convex optimization settings.

Finite population evolutionary models over time evolving fitness landscapes are typically studied via simulations (e.g., [16] and references therein). These models have also inspired evolutionary models of computation, e.g., genetic algorithms, whose study under dynamic fitness environments is a well established area with many applications (e.g., [36] and references therein) but with little theoretical understanding and even theoretical papers on the subject typically rely on combinations of analytical and experimental results [5].

3 Preliminaries

Notation. All vectors are in bold-face letters, and are considered as column vectors. To denote a row vector we use $\mathbf{x}^\top$. The i-th coordinate of $\mathbf{x}$ is denoted by x_i. Let $\Delta_n = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} \geq 0,\ \sum_{i=1}^n x_i = 1\}$ be the set of probability distributions on n coordinates. For given matrix A define $A_{\max}, A_{\min}$ the largest, smallest entry in matrix A respectively and $(A\mathbf{x})_i := \sum_j A_{ij}x_j$. Define $\mathrm{Supp}(\mathbf{x}) = \{i \mid x_i \neq 0\}$.

3.1 Dynamics: Discrete Replicator Dynamics with/without Mutation

For a haploid species[2] (one with single set of chromosomes, unlike diploids such as humans who have chromosome pairs) with two genes (coordinates), let S_1 and S_2 be the set of possible alleles (types) for the first and second gene respectively. Then, an individual of such a species can be represented by an ordered pair $(i, j) \in S_1 \times S_2$. Let W_{ij} be the fitness of such an individual capturing its ability to reproduce during a mating. Thus, fitness landscape of such a species can be represented by matrix W of dimension $n \times n$, where we assume that $n = |S_1| = |S_2|$.

Sexual Model without mutation. In every generation, each individual (i, j) mates with another individual (i', j') picked uniformly at random from the population (can pick itself). The offspring can have any of the four possible combinations, namely (i, j), (i, j'), (i', j), (i', j'), with equal probability. Let x_i be a random variable that denotes the proportion of the population with allele i in the first coordinate, and similarly y_j be the frequency of the population with allele j in the second coordinate. After one generation, the expected number of offsprings with allele i in first coordinate is proportional to $x_i \cdot x_i \cdot (W\mathbf{y})_i + 2\frac{1}{2}(1 - x_i)x_i \cdot (W\mathbf{y})_i = x_i(W\mathbf{y})_i$ (x_i^2 stands for the probability both individuals have allele i in the first coordinate - which the offspring will inherit - and $2\frac{1}{2}(1 - x_i)x_i$ stands for the probability that exactly one of the individuals has allele i in the first coordinate and the offspring will inherit). Similarly the expected number of offsprings with allele j for the second coordinate is $y_j(W^\top \mathbf{x})_j$. Hence, if $\mathbf{x}', \mathbf{y}'$ denote the frequencies of the alleles in the population in the next generation (random variables)

$$E[x_i'|\mathbf{x}, \mathbf{y}] = \frac{x_i(W\mathbf{y})_i}{\mathbf{x}^\top W\mathbf{y}} \text{ and } E[y_j'|\mathbf{x}, \mathbf{y}] = \frac{y_j(W^\top \mathbf{x})_j}{\mathbf{x}^\top W\mathbf{y}}.$$

We are interested in analyzing a *deterministic* version of the equations above, which essentially captures an infinite population model. Thus if frequencies at time t are denoted by $(\mathbf{x}(t), \mathbf{y}(t))$, they obey the following dynamics governed by the function $g : \Delta \to \Delta$, where $\Delta = \Delta_n \times \Delta_n$:

$$\text{Let } (\mathbf{x}(t+1), \mathbf{y}(t+1)) = g(\mathbf{x}(t), \mathbf{y}(t)), \text{ where } \begin{array}{l} \forall i \in S_1,\ x_i(t+1) = x_i(t)\frac{(W\mathbf{y}(t))_i}{\mathbf{x}^\top(t)W\mathbf{y}(t)} \\ \forall j \in S_2,\ y_j(t+1) = y_j(t)\frac{(W^\top\mathbf{x}(t))_j}{\mathbf{x}^\top(t)W\mathbf{y}(t)}. \end{array} \quad (1)$$

It is easy to see that g is well-defined when W is a positive matrix. Chastain et al. [8] gave a game theoretic interpretation of the deterministic equations (1). It can be seen as a repeated two player coordination game (each gene is a player), the possible alleles for a gene are its pure strategies and both players play according to dynamics (1). A modification of these dynamics has also appeared in models of grammar acquisition [27], and can be seen as the discrete analogue of continuous replicator dynamics [19]. Furthermore, Mehta et al. [22] showed that dynamics with equations (1) converges point-wise to a pure fixed point, i.e., where exactly one coordinate is non-zero in both $\mathbf{x}$ and $\mathbf{y}$, for all but measure zero of initial conditions in Δ, when W has distinct entries.

Sexual Model with mutation. Next we extend the dynamics of (1) to incorporate mutation. The mutation model which appears in Hofbauer's book [13], is a two step process. The first step is governed by (1), and after that in each individual, and for each of its gene,

[2] See Section A in the appendix for a short discussion of all relevant biological terms.

corresponding allele, say k, mutates to another allele of the same gene, say k', with probability $\tau > 0$ for all $k' \neq k$. After a simple calculation (see C.8 for calculations) the resulting dynamics turns out to be as follows, where f is a $\Delta \to \Delta$ function:

$$\text{Let } (\mathbf{x}(t+1), \mathbf{y}(t+1)) = f(\mathbf{x}(t), \mathbf{y}(t)), \text{ then } \begin{array}{l} x_i(t+1) = (1-n\tau)x_i(t)\frac{(W\mathbf{y}(t))_i}{\mathbf{x}(t)^\top W\mathbf{y}(t)} + \tau, \\ y_j(t+1) = (1-n\tau)y_j(t)\frac{(\mathbf{x}(t)^\top W)_j}{\mathbf{x}(t)^\top W\mathbf{y}(t)} + \tau. \end{array} \tag{2}$$

3.2 Our model

In this paper we will analyze a noisy version of (1), (2). Essentially we add small random noise to non-zero coordinates of $(\mathbf{x}(t), \mathbf{y}(t))$ [3].

▶ **Definition 1.** Given $\mathbf{z} \in \Delta$ and a small $0 < \delta$ (δ is $o_n(\tau)$), define $\Delta(\mathbf{z}, \delta)$ to be a set of vectors $\{\mathbf{z} + \delta \in \Delta \mid \text{Supp}(\delta) = \text{Supp}(\mathbf{z});\ \delta_i \in \{-\delta, +\delta\},\ \forall i\}$.[4]

Note that if $\mathbf{z}$ is pure (has support size one), then δ is all zero vector[5]. Define noisy versions of both g from (1) and f from (2) as follows: Given $(\mathbf{x}(t), \mathbf{y}(t))$ pick $\delta_\mathbf{x} \in \Delta(\mathbf{x}(t), \delta)$ and $\delta_\mathbf{y} \in \Delta(\mathbf{y}(t), \delta)$ uniformly at random. Set with probability half $\delta_\mathbf{x}$ to zero, and with the other half set $\delta_\mathbf{y}$ to zero. Then redefine dynamics g of (1) as follows:

$$(\mathbf{x}(t+1), \mathbf{y}(t+1)) = g_\delta(\mathbf{x}(t), \mathbf{y}(t)) = g(\mathbf{x}(t), \mathbf{y}(t)) + (\delta_\mathbf{x}, \delta_\mathbf{y}). \tag{3}$$

And redefine dynamics f of (12) capturing sexual evolution with mutation as follows.

$$(\mathbf{x}(t+1), \mathbf{y}(t+1)) = f_\delta(\mathbf{x}(t), \mathbf{y}(t)) = f(\mathbf{x}(t), \mathbf{y}(t)) + (\delta_\mathbf{x}, \delta_\mathbf{y}). \tag{4}$$

Furthermore, we will have that if any x_i, y_j goes below δ, we set it to zero. This is crucial for our theorems because otherwise the dynamics with equations (1) and (2) (or even (3) and (4)) can converge to a fixed point at $t \to \infty$, but never reach a point in a finite amount of time. This is true in the result of [22], the dynamics converge almost surely to pure fixed points as $t \to \infty$ but do not reach fixation in a finite time. So x_i, y_j reaches fixation (set it to zero) if $x_i, y_j < \delta$. We need to re-normalize after this step.

$$\begin{array}{l} \forall i \in S_1, \text{ if } x_i(t) < \delta \text{ then set } x_i(t) = 0. \text{ Re-normalize } \mathbf{x}(t). \\ \forall j \in S_1, \text{ if } y_j(t) < \delta \text{ then set } y_j(t) = 0. \text{ Re-normalize } \mathbf{y}(t). \end{array} \tag{5}$$

▶ **Definition 2.** We call a vector $\mathbf{v}$ *negligible* if there exists an i s.t $v_i < \delta$.

Tracking population size. Suppose the size of the initial population is N^0, and let population at time t be N^t. In every time period N^t gets multiplied by average fitness of the current population, namely $\mathbf{x}(t)^T W(t)\mathbf{y}(t)$, where $(\mathbf{x}(t), \mathbf{y}(t))$ denote the frequencies of alleles at generation t and $W(t)$ the matrix fitness/environment at time (see discussion below about changing of environments).

$$\text{Let average fitness } \Phi^t = \mathbf{x}(t)^\top W(t)\mathbf{y}(t) \quad \text{then } \mathbb{E}[N^{t+1} | \mathbf{x}(t), \mathbf{y}(t), N^t] = N^t\Phi^{t+1} \tag{6}$$

We will consider $N^{t+1} = N^t\Phi^{t+1}$ (this is not necessarily an integer at every step; it captures the average population size, see also [33]). Based on the value of N^t, we give the definition of survival and extinction.

[3] This is different from diffusion approximation, noise helps to avoid saddle points essentially.
[4] In case the size of the support of $\mathbf{z}$ is odd, there will be a zero entry in δ, so $|\text{Supp}(\delta)| = |\text{Supp}(\mathbf{z})| - 1$
[5] There are no sampling errors in monomorphic population

▶ **Definition 3.** We say the population *goes extinct* if for initial population size N^0, there exists a time t so that $N^t < 1$. On the other hand, we say that population *survives* if for all times $t \in \mathbb{N}$ we have that $N^t \geq 1$. Another definition one can have is that the population goes extinct if $\liminf_{t\to\infty} N^t = 0$, and survives if $\liminf_{t\to\infty} N^t > 0$. For the rest of the paper we use the former definition, but our results and arguments work for the latter definition as well.

3.2.1 Model of environment change

Following the work of Wolf et al. [35], we consider a Markov chain based model of changing environment. Let $\mathcal{E}$ be the set of different possible environments, and W^e be the fitness matrix in environment $e \in \mathcal{E}$. E denotes the set of (e, e') pairs if there is a non-zero probability $p_{e,e'} \in (0, 1)$ to go from environment e to e'. See Figure 1 for an example. For a parameter $p < 1$ we assume that $\sum_{e':(e,e')\in E} p_{e,e'} \leq p,\ \forall e \in \mathcal{E}$. That is, after every generation of the dynamics (3) or (4), the environment changes to one of its neighboring environment with probability at most $p < 1$, and remains unchanged with probability at least $(1 - p)$. The graph formed by edges in $\mathcal{E}$ is assumed to be connected, thus the resulting Markov chain eventually will stabilize to a stationary distribution π_e (is ergodic).

Even though fitness matrices W^e can be arbitrary, it is generally assumed that W^e has distinct positive entries [7, 22]. Furthermore, no individual can survive all the environments on average. Mathematically, if π_e is the stationary distribution of this Markov chain then, $\forall i, j, \quad \prod_{e\in\mathcal{E}} (W^e_{ij})^{\pi_e} < 1$. Furthermore, we assume that every environment has alleles of good type as well as bad type. An allele i of good type has uniform fitness (i.e., $\frac{\sum_j W_{ij}}{n}$) of at least $(1 + \beta)$ for some $\beta > 0$, and alleles of bad type are dominated by a good type allele point-wise.[6] Finally, the number of bad alleles are $o(n)$ (sublinear in n). Let the set of bad alleles for genes $i = 1, 2$ in environment e be denoted by B^e_i.

Putting all of the above together, the Markov chain for environment change is defined by set $\mathcal{E}$ of environments and its adjacency graph, fitness matrices W^e, $\forall e \in \mathcal{E}$, probability $1 - p$ with which dynamics remains in current environment, sets $B^e_i \subset S_i,\ i = 1, 2$ of bad alleles in environment e, and $\beta > 0$ to lower-bound average fitness of good type alleles. See also Section B.2 for discussion on the assumptions where we claim that most of them are necessary for our theorems. In the next sections we will analyze the dynamics with equations (3, 4) in terms of convergence and population size for fixed and dynamic environments.

4 Overview of proofs

The dynamical systems that we analyze, namely (3) and (4), under the evolving environment model of Section 3.2.1 are (stochastically perturbed) nonlinear replicator-like dynamical systems whose parameters evolve according to a (possibly slow mixing) Markov chain. We reduce the analysis of this complex setting to a series of smaller, modular arguments that combine as set-pieces to produce our main theorems.

Convergence rate for evolution without mutation in static environment. Our starting point is [22] where it was shown that in the case of noise-free sexual dynamics governed by (1) the average population fitness increases in each step and the system converges to equilibria, and moreover that for almost all initial conditions the resulting fixed point corresponds to a

[6] Think of bad type alleles akin to a terminal genetic illness. Such assumptions are typical in the biological literature (e.g., [16]).

Table 1 List of parameters.

Symbol	Interpretation
W^e	fitness matrix at environment e
$W(t), W^{e(t)}$	fitness matrix at time t
γ^e	minimum difference between entries in fitness matrix W^e
$\mathbf{x}, \mathbf{y}$	frequencies of (alleles) strategies
δ	noise/perturbation
Φ	potential/average fitness $\mathbf{x}^\top W \mathbf{y}$
β	If allele i is of good type in environment e then it satisfies $\frac{\sum_j W^e_{ij}}{n} \geq 1+\beta$
τ	probability that an individual with allele k mutates to k' (of the same gene)

monomorphic population (pure/not mixed equilibrium). Conceptually, the first step in our analysis tries to capitalize on this stronger characterization by showing that convergence to such states happens fast. This is critical because while there are only linearly many pure equilibria, there are (generically) exponentially many isolated, mixed ones [7], which are impossible to meaningfully characterize. By establishing the predictive power of pure states, we radically reduce our uncertainty about system behavior and produce a building block for future arguments.

Without noise we cannot hope to prove fast convergence to pure states since by choosing initial conditions sufficiently close to the stable manifold of an unstable equilibrium, we are bound to spend super-polynomial time near such unstable states. In finite population models, however, the system state (proportions of different alleles) is always subject to small stochastic shocks (akin to sampling errors). These small shocks suffice to argue fast convergence by combining an inductive argument and a potential/Lyapunov function argument.

To bound the convergence time to a pure fixed point starting at an arbitrary mixed strategy (maybe with full support), it suffices to bound the time it takes to reduce the size of the support by one, because once a strategy x_i becomes zero it remains zero under (3), i.e., an extinct allele can never come back in absence of mutations (and then use induction). For the inductive step, we need two non trivial arguments. First we need a lower bound on the rate of increase of the mean population fitness when the dynamics is not at approximate fixed points[7], shown in Lemma 4. This requires a quantitative strengthening of potential/(nonlinear dynamical system) arguments in [22]. Secondly, we show that the noise suffices to escape fast (with high probability) from the influence of fixed points that are not monomorphic (these are like saddle points). We used a combination of stochastic techniques including origin returning random walks, Azuma type inequalities for submartingales, and arguing about the increase in expected mean fitness $\mathbf{x}(t)^\top W(t)\mathbf{y}(t)$ in a few steps (Lemmas 7-11), where $\mathbf{x}$ and $\mathbf{y}$ capture allele frequencies at time step t. As a result we show polynomial-time convergence of (3) to pure equilibrium under static environment in Theorem 12. This result may be of independent interest since fast convergence of nonlinear dynamics to equilibrium is not typical [25].

Survival, extinction under dynamic environments. As described in Section 3.2.1, we consider a Markov chain based model of environmental changes, where after every selection step, the fitness matrix changes with probability at most p. Suppose the starting population

[7] We call these states α-close points.

size is $N^0 > 0$ and let N^t denote the size at time t then in every step N^t gets multiplied by the mean fitness $\mathbf{x}(t)^\top W(t)\mathbf{y}(t)$ of the current population (see (6)). We say that population goes extinct if for some t, $N^t < 1$, and it survives if $N^t \geq 1$, for all t.

We assume that there do not exist "all-weather" phenotypes. We encode this by having the monomorphic population of any genotype decrease when matched to an environment chosen according to the stationary distribution of the Markov chain.[8] In other words, an allele may be both "good" and "bad" as environment changes, sometimes leading to growth, and other times to decrease in population.

Case a) sexual evolution without mutation. If the population becomes monomorphic then this single phenotype can not survive in all environments, and will eventually wither as its population will be in exponential decline once the Markov chain mixes. The question is whether monomorphism is achieved under changing environment; the above analysis is not applicable directly as the fitness matrix is not fixed any more. Our first theorem (Theorem 12) upper bounds the amount of time T needed to "wait" in a single environment so as the probability of convergence to a monomorphic state is at least some constant (e.g., $\frac{1}{2}$). Breaking up the time history in consecutive chunks of size T and applying Borel-Cantelli theorem implies that the population will become monomorphic with probability one (Theorem 14). This is the strongest possible result without explicit knowledge of the specifics of the Markov chain (e.g., mixing time).

Case b) sexual evolution with mutation. As described in Section 3.1, we consider a well-established model of mutation [13], where after every selection step, each allele mutates to another with probability τ. The resulting dynamics is governed by (2), and we analyze its noisy counterpart (4). This ensures that in each period the proportion of every allele is at least τ. We show that this helps the population survive.

Unlike the *no mutation* case [22], the average fitness $\mathbf{x}(t)^\top W\mathbf{y}(t)$ is no more increasing in every step, even in absence of noise. Instead we derive another potential function that is a combination of average fitness and entropy. Due to mutations forcing exploration, natural selection weeds out the bad alleles fast (Lemma 15). Thus there may be initial decrease in fitness, however the decrease is upper bounded. Furthermore, we show that the fitness is bound to increase significantly within a short time horizon due to increase in population of good alleles (Lemma 16). Since population size gets multiplied by average fitness in each iteration, this defines a biased random walk on logarithm of the population size. Using upper and lower bounds on decrease and increase respectively, we show that the probability of extinction stochastically dominates a simpler to analyze random variable pertaining to biased random walks on the real line (Lemma 17). Thus, the probability of long term survival is strictly positive (Theorem 18). This completes the proof of informal Theorem 1.

Deterministic convergence despite mutation in static environments. Finally, as an independent result for the case of noise free dynamics (infinite population) with mutation governed by (2), we show convergence to fixed points in the limit, by defining a novel potential function which is the product of mean fitness $\mathbf{x}^\top W\mathbf{y}$ and a term capturing diversity of the allele distribution (Theorem 20). The latter term is essentially the product of allele

[8] If, for any genotype, the population increased in expectation over the randomly chosen environment, then once monomorphic population consisting of only such a genotype is reached, the population would blow up exponentially (and forever) as soon as the Markov chain reached its mixing time.

frequencies ($\prod_i x_i \prod_i y_i$). Such convergence results are not typical in dynamical systems literature [25], and therefore this potential function may be useful to understand limit points of this and similar dynamics (the continuous time analogue can be found here [13]). One way to interpret this result is a homotopy method for computing equilibria in coordination games, where the algorithm always converges to fixed points, and as mutation goes to zero the stable fixed points correspond to the pure Nash equilibria [7].

5 Rate of Convergence: Dynamics without Mutation in Fixed Environments

In this section we show a polynomial bound on the convergence time of dynamics (3), governing sexual evolution under natural selection with noise, in a static environment. In addition, we show that the fixed points reached by the dynamics are pure.

Consider a fixed environment e and we use W to denote its fitness matrix W^e. It is known that average fitness $\mathbf{x}^\top W\mathbf{y}$ increases under the non-noisy counterpart (1) [22]. In the next lemma we obtain a lower bound on this increase.

▶ **Lemma 4.** *Let $(\hat{\mathbf{x}},\hat{\mathbf{y}}) = g(\mathbf{x},\mathbf{y})$ where $(\mathbf{x},\mathbf{y}) \in \Delta$ and g is from equation (1). Then,*

$$\hat{\mathbf{x}}^\top W\hat{\mathbf{y}} - \mathbf{x}^\top W\mathbf{y} \geq C\left(\sum_i x_i\left((W\mathbf{y})_i - \mathbf{x}^\top W\mathbf{y}\right)^2 + \sum_i y_i\left((W^\top\mathbf{x})_i - \mathbf{x}^\top W\mathbf{y}\right)^2\right)$$

for $C = \frac{3}{8\cdot\max_{i,j} W_{ij}}$.

For the rest of the section, C denotes $\frac{3}{8\cdot W_{\max}}$ where $W_{\max} = \max_{ij} W_{ij}$ and $W_{\min} = \min_{ij} W_{ij}$. Note that the lower bound obtained in Lemma 4 is strictly positive unless $(\mathbf{x},\mathbf{y})$ is a fixed point of (1). This gives an alternate proof of the fact that, under dynamics (1), average fitness is a potential function, i.e., increases in every step. On the other hand, the lower bound can be arbitrarily small at some points, and therefore it does not suffice to bound the convergence time. Next, we define points where this lower-bound is relatively small.

▶ **Definition 5.** We call a point $(\mathbf{x},\mathbf{y})$ α-close for an $\alpha > 0$, if for all $\mathbf{x}',\mathbf{y}' \in \Delta$ such that $\mathrm{Supp}(\mathbf{x}') \subseteq \mathrm{Supp}(\mathbf{x})$ and $\mathrm{Supp}(\mathbf{y}') \subseteq \mathrm{Supp}(\mathbf{y})$ we have $|\mathbf{x}^\top W\mathbf{y} - \mathbf{x}'^\top W\mathbf{y}| \leq \alpha$ and $|\mathbf{x}^\top W\mathbf{y} - \mathbf{x}^\top W\mathbf{y}'| \leq \alpha$.

α-close points, are a specific class of *"approximate" stationary points*, where the progress in average fitness is not significant (see Figure 3, the big circles contain these points). From now on, think α as a small parameter that will be determined in the end of this section. If a given point $(\mathbf{x},\mathbf{y})$ is not α-close and not *negligible* (see Definition 2) then using Lemma 4 it follows that the increase in potential is at least $C\delta\alpha^2$. Formally:

▶ **Corollary 6.** *If $(\mathbf{x},\mathbf{y}) \in \Delta$ is neither α-close nor negligible, and $(\hat{\mathbf{x}},\hat{\mathbf{y}}) = g(\mathbf{x},\mathbf{y})$, then*

$$\hat{\mathbf{x}}^\top W\hat{\mathbf{y}} \geq \mathbf{x}^\top W\mathbf{y} + C\delta\alpha^2.$$

Proof. Since the vector $(\mathbf{x},\mathbf{y})$ is neither α-close nor negligible, it follows that there exists an index i such that $|(W\mathbf{y})_i - \mathbf{x}^\top W\mathbf{y}| > \alpha$ and $x_i \geq \delta$ and hence $x_i((W\mathbf{y})_i - \mathbf{x}^\top W\mathbf{y})^2 > \delta\alpha^2$, or $|(W^\top\mathbf{x})_i - \mathbf{x}^\top W\mathbf{y}| > \alpha$ and $y_i \geq \delta$ and hence $y_i((W^\top\mathbf{x})_i - \mathbf{x}^\top W\mathbf{y})^2 > \delta\alpha^2$. Therefore in Lemma 4, the r.h.s is at least $C\delta\alpha^2$ and thus we get that $\hat{\mathbf{x}}^\top W\hat{\mathbf{y}} - \mathbf{x}^\top W\mathbf{y} \geq C\delta\alpha^2$. ◀

In the analysis above we considered non-noisy dynamics governed by (1). Our goal is to analyze finite population dynamics, which introduces noise and the resulting dynamics is (3). This changes how the fitness increases/decreases. The next lemma shows that in expectation the average fitness remains unchanged after the introduction of noise.

▶ **Lemma 7.** *Let $\delta = (\delta_{\mathbf{x}}, \delta_{\mathbf{y}})$ be the noise vector. It holds that $\mathbb{E}_\delta[(\mathbf{x}+\delta_{\mathbf{x}})^\top W(\mathbf{y}+\delta_{\mathbf{y}})] = \mathbf{x}^\top W\mathbf{y}$.*

Next, we show how random noise can help the dynamic escape from a polytope of α-close points. We first analyze how adding noise may help increase fitness with high enough probability. A simple application of Catalan numbers shows that:

▶ **Lemma 8.** *The probability of a (unbiased) random walk on the integers that consist of $2m$ steps of unit length, beginning at the origin and ending at the origin, that never becomes negative is $\frac{1}{m+1}$.*

We define $\gamma = \min_{(i,j)\neq(i',j')} |W_{ij} - W_{i'j'}|$. The following lemma is essentially a corollary of Lemma 8.

▶ **Lemma 9.** *Let $\delta_{\mathbf{y}}$ be a random noise with support size m. For all i in the support of $\mathbf{x}$ we have that $(W\delta_{\mathbf{y}})_i \geq \frac{\gamma\delta m}{2}$ with probability at least $\frac{1}{1+m/2}$ (same is true for $\delta_{\mathbf{x}}$ and $\mathbf{y}$).*

We will also need the following theorem due to Azuma [10] on submartingales.

▶ **Theorem 10** (Azuma [10])**.** *. Suppose $\{X_k, k = 0, 1, 2, ..., N\}$ is a submartingale and also $|X_k - X_{k-1}| < c$ almost surely then for all positive integers N and all $t > 0$ we have that*

$$\mathbb{P}[X_N - X_0 \leq -t] \leq e^{-\frac{t^2}{2Nc^2}}$$

Towards our main goal of showing polynomial time convergence of the noisy dynamics (3) (shown in Theorem 12), we need to show that the fitness increases within a few iterations of the dynamic with high probability. It suffices to show that the average fitness under some transformation is a submartingale, and then the result will follow using Azuma's inequality.

▶ **Lemma 11.** *Let Φ^t be the random variable which corresponds to the average fitness at time t. Assume that for the time interval $t = 0, ..., 2T$ the trajectory $(\mathbf{x}(t), \mathbf{y}(t))$ has the same support. Let $m = \max\{|Supp(\mathbf{x}(t))|, |Supp(\mathbf{y}(t))|\}$, and the non-zero entries of $(\mathbf{x}(t), \mathbf{y}(t))$ be at least δ. If $\frac{1}{(m+2)}(\frac{\gamma\delta m}{2} - 2\alpha)^2 \geq \delta\alpha^2$ then we have that*

$$E[\Phi^{2t+2}|\Phi^{2t}, ..., \Phi^0] \geq \Phi^{2t} + C\delta\alpha^2.$$

In other words, the sequence $Z^t \equiv \Phi^{2t} - t \cdot C\delta\alpha^2$ for $t = 1, ..., T$ is a submartingale and also $|Z^{t+1} - Z^t| \leq W_{\max} - W_{\min}$.

Using all the above analysis and Azuma's inequality (Theorem 10), we establish our first main result on convergence time of the noisy dynamics governed by (3) for sexual evolution under natural selection and without mutation.

▶ **Theorem 12** (Main 2)**.** *For all initial conditions $(\mathbf{x}(0), \mathbf{y}(0)) \in \Delta$, the dynamics governed by (3) in an environment represented by fitness matrix W reaches a pure fixed point with probability $1 - \epsilon$ after $O\left(\frac{(W_{\max})^4 n \ln(\frac{2n}{\epsilon})}{\delta^6\gamma^4}\right)$ iterations.*

Proof. It suffices to show that support size of the $\mathbf{x}$ or $\mathbf{y}$ reduces by one in a bounded number of iterations with at least $1 - \frac{\epsilon}{2n}$ probability.

Using Lemma 11 we have that the random variable $\Phi^{2t} - t \cdot C\delta\alpha^2$ is a submartingale and since $W_{\min} \leq \Phi^t \leq W_{\max}$ we use Azuma's inequality 10 and we get that

$$\mathbb{P}[\Phi^{2t} - t \cdot C\delta\alpha^2 \leq \Phi^0 - \lambda] \leq e^{-\frac{\lambda^2}{2tW_{max}^2}},$$

hence for $\lambda = \sqrt{2tW_{max}^2 \ln(\frac{2n}{\epsilon})}$ we get that the average fitness after $2t$ steps will be at least $\Phi^0 - \sqrt{2tW_{max}^2 \ln(\frac{2n}{\epsilon})} + t \cdot C\delta\alpha^2$ with probability at least $1 - \frac{\epsilon}{2n}$. By setting $t \geq \frac{8W_{\max}^2}{C^2\delta^2\alpha^4} \ln\left(\frac{2n}{\epsilon}\right)$ we have that the average fitness at time $2t$ will be greater than $W_{\max}$ with probability $1 - \frac{\epsilon}{2n}$, but since the potential is at most $W_{\max}$ for all vectors in the simplex, it follows that at some point the frequency vector become *negligible*, i.e., a coordinate of $\mathbf{x}$ or $\mathbf{y}$ became less than δ. Hence, the probability that the support size decreased during the process is at least $1 - \frac{\epsilon}{2n}$.

By union bound (the initial support size is at most $2n$) we conclude that dynamics (3) reaches a pure fixed point with probability $1 - \epsilon$ after t iterations with $t = 2n\frac{8W_{\max}^2}{C^2\delta^2\alpha^4} \ln\left(\frac{2n}{\epsilon}\right)$. Finally, for assumption $\frac{1}{(m+2)}(\frac{\gamma\delta m}{2} - 2\alpha)^2 \geq \delta\alpha^2$ used in Lemma 11 to hold for $2 \leq m \leq n$, we set α to be such that $\alpha \leq \frac{\gamma\delta}{4}$ where we have $\frac{4(m-1)^2}{(m+2)} \geq 1 > \delta$. Using such an α it follows that dynamics (3) reaches a pure fixed point with probability $1 - \epsilon$ after $\frac{2^{18}}{9} \times \frac{nW_{\max}^4}{\delta^6\gamma^4} \ln\left(\frac{2n}{\epsilon}\right)$ iterations. ◀

6 Changing Environment: Survival or Extinction?

In this section we analyze how evolutionary pressures under changing environment may lead to survival/extinction depending on the underlying mutation level. Motivated from Wolf et al. work [35], we use Markov chain based model to capture the changing environment, where every state captures a particular environment (see Section 3.2.1 for details).

6.1 Extinction without mutation

We show that the population goes extinct with probability one, if the evolution is governed by (3), i.e., natural selection *without* mutations under sexual reproduction. The proof of this result critically relies on polynomial-time convergence to monomorphic population shown in Theorem 12 in case of fixed environment.

As discussed in Section 3.2.1, we have assume that the Markov chain is such that no individual can be fit to survive in all environments. Formally,

$$\forall i, j, \quad \prod_{e \in \mathcal{E}} (W_{ij}^e)^{\pi_e} < 1. \tag{7}$$

Thus, if we can show convergence to monomorphic population under evolving environments as well then the extinction is guaranteed using (7) and the fact that population size N^t gets multiplied by current average fitness (see (6)). However, showing convergence in stochastically changing environment is tricky because environment can change in any step with some probability and then the argument described in the previous section breaks down. To circumvent this we will make use of Borel-Cantelli theorem where we say that *an event happens* if environment remains unchanged for a large but fixed number of steps.

▶ **Theorem 13** (Second Borel-Cantelli [11]). *Let $E_1, E_2, \ldots$ be a sequence of events. If the events E_n are independent and the sum of the probabilities of the E_n diverges to infinity, then the probability that infinitely many of them occur is 1.*

Using the above theorem with appropriate *event* definition, we prove the first part of the main result stated in Theorem 1.

▶ **Theorem 14** (Main 1a). *Regardless of the initial distributions $(\mathbf{x}(0), \mathbf{y}(0)) \in \Delta$, the population goes extinct with probability one under dynamics governed by (3), capturing sexual evolution without mutation under natural selection.*

Proof. Let T^e be the number of iterations the dynamics (3) need to reach a pure fixed point with probability $\frac{1}{2}$. Theorem 12 implies $T^e = O\left(\frac{nW_{\max}^{e\,4}}{\delta^6\gamma^{e\,4}}\ln 4n\right)$. Let $T = \max_e T^e$. We consider the time intervals $1, ..., T,\ T+1, ..., 2T, ...$ which are multiples of T. The probability that Markov chain will remain at a specific environment e in the time interval $kT+1, ..., (k+1)T$ is $\rho_k = (1-p)^T$. We define the sequence of events $E_1, E_2, ...$, where E_i corresponds to the fact that the chain remains in the same environment from time $(i-1)T+1, ..., iT$. It is clear that E_i's are independent and also $\sum_{i=1}^{\infty}\mathbb{P}[E_i] = \sum_{i=1}^{\infty}\rho_i = \infty$. From Borel-Cantelli Theorem 13 it follows that E_i's happen infinitely often with probability 1. When E_i happens there is a time interval of length T that the chain remains in the same environment and therefore with probability $\frac{1}{2}$ the dynamics will reach a pure fixed point. After E_i happen for k times, the probability to reach a pure fixed point is at least $1-\frac{1}{2^k}$. Hence with probability one (letting $k\to\infty$), the dynamics (3) will eventually reach a pure fixed point.

To finish the proof, let T_{pure} be a random variable that captures the time when a pure fixed point, say (i,j), is reached. The population will have size at most $N^0V^{T_{pure}}$ where $V = \max_e W_{\max}^e$. Under the assumption on the entries (see inequality (7)) it follows that at any time T' sufficiently large we get that the population at time $T'+T_{pure}$ will be roughly at most

$$N^0V^{T_{pure}}\prod_e (W_{ij}^e)^{T'\pi_e} = N^0V^{T_{pure}}\left(\prod_e (W_{ij}^e)^{\pi_e}\right)^{T'}.$$

By choosing $T' \geq \frac{\ln(N^0V^{T_{pure}})}{-\ln\left((W_{ij}^e)^{\pi_e}\right)}$ (and also satisfying the constraint that is much greater than the mixing time) it follows that $N^{T'+T_{pure}} < 1$ and hence the population dies. So, the population goes extinct with probability one in the dynamics without mutation. ◀

6.2 Survival with mutation

In this section we consider evolutionary dynamics governed by (4) capturing sexual evolution *with* mutation under natural selection. Contrary to the case where there are no mutations we show that population survives with positive probability. Furthermore, this result turns out to be robust in the sense that it holds even when every environment has some (few) very bad type alleles. Also, the result is independent of the starting distribution of the population.

The main intuition behind proving this result is that, as for the mutation model in [13], every allele is carried by at least τ fraction of the population in every generation. Therefore even if a"good" allele becomes "bad" as the environment changes, as far as the new environment has a few fit alleles, there will be some individuals carrying those who will then procreate fast, spreading their alleles further and leading to overall survival. However, unlike in the no mutation case [22], average fitness is no more a potential function even for non-noisy dynamics, i.e., it may decrease, and therefore showing such an improvement is tricky.

First we show that if some small amount of time is spent in an environment then the frequencies of the bad alleles become small and their effect is negligible, independent of the population distribution at the time when this environment was entered. Recall the assumption on good/bad type alleles (Section 3.2.1). Formally, let B_i^e be the set of bad type alleles for $i = 1, 2$ in environment e,

$$\begin{aligned} &\forall i \in S_1 \setminus B_1^e,\ \frac{\sum_j W_{ij}^e}{n} \geq 1+\beta, \quad \text{and} \quad \forall i \in S_1 \setminus B_1^e, \forall k \in B_1^e, W_{ij}^e \geq W_{kj}^e,\ \forall j \\ &\forall j \in S_2 \setminus B_2^e,\ \frac{\sum_i W_{ij}^e}{n} \geq 1+\beta, \quad \text{and} \quad \forall j \in S_2 \setminus B_2^e, \forall k \in B_2^e, W_{ij}^e \geq W_{ik}^e,\ \forall i \end{aligned} \tag{8}$$

▶ **Lemma 15.** *Suppose that the environment e is static for time at least $t \geq \frac{\ln(2n)}{n\tau}$. For any $(\mathbf{x}(0), \mathbf{y}(0)) \in \Delta$, we have that $\sum_{i \in B_1^e} x_i(t) + \sum_{j \in B_2^e} y_i(t) \leq \frac{2(|B_1^e|+|B_2^e|)}{n} = \frac{2|B^e|}{n}$ with $B^e = B_1^e \cup B_2^e$.*

Using the fact that number of individuals with bad type alleles decreases very fast, established in Lemma 15, we can prove that within an environment while there may be decrease in average fitness initially, this decrease is lower bounded. Moreover, it will later increase fast enough so that the initial decrease is compensated.

▶ **Lemma 16.** *Suppose that the environment e is static for time t and also $\tau \leq \frac{\beta}{16n}$, $|B^e| \ll n\beta$ then there exists a threshold time T_{thr} such that for any given initial distributions of the alleles $(\mathbf{x}(0), \mathbf{y}(0)) \in \Delta$, if $t < T_{thr}$ then the population size will experience a loss factor of at most $\frac{1}{d}$, otherwise it will experience a gain factor of at least d for some $d > 1$, where $T_{thr} = \frac{6\ln(2n)}{n\tau\beta W_{\min}}$ and $W_{\min} = \min_e W_{\min}^e$.*

To show the second part of Theorem 1 (main result), we will couple the random variable corresponding to the number of individuals at every iteration with a biased random walk on the real line. This can be done since in Lemma 16 we established that the decrease and increase in average fitness is upper and lower bounded, respectively. We will apply the following well-known lemma about the biased random walks.

▶ **Lemma 17** (Biased random walk)**.** *Assume we perform a random walk on the real line, starting from point $k \in \mathbb{N}$ and going right (+1) with probability $q > \frac{1}{2}$ and left (-1) with probability $1 - q$. The probability that we will eventually reach 0 is $\left(\frac{1-q}{q}\right)^k$.*

Using Lemma 16 together with the biased random walk Lemma 17, we show our next main result on survival of population under mutation in the following theorem.

▶ **Theorem 18** (Main 1b)**.** *If $p < \frac{1}{2T_{thr}}$ where $T_{thr} = \frac{6\ln(2n)}{n\tau\beta W_{\min}}$ then the probability of survival is at least $1 - \left(\frac{pT_{thr}}{1-pT_{thr}}\right)^{c \ln N^0}$ for some c independent of N^0, $c = \left(\frac{n\tau W_{\min}}{\ln(2n)}\right)$.*

Proof. The probability that the chain remains at a specific environment for least T_{thr} iterations is $(1-p)^{T_{thr}} > 1 - pT_{thr}$ (from the moment it enters the environment until it departs) and hence the probability that the chain stays at an environment for time less that T_{thr} is at most pT_{thr}. Let $N^t = N^0 \prod_{j=1}^t \mathbf{x}(j)^\top W^{e(j)} \mathbf{y}(j)$ (see (6) where here $e(j)$ corresponds to the environment at time j) the number of individuals at time t and Z^i be the position of the biased random walk at time i as defined in Lemma 17 with $q = 1 - pT_{thr}$ and assume that $Z^0 = \lfloor \log_d N^0 \rfloor$ (d is from lemma 16). Let $t_1, t_2, \ldots$ be the sequence of times where there is a change of environment (with $t_0 = 0$) and consider the trivial coupling where when the chain changes environment then a move is made on the real line. If the chain remained in the environment for time less than T_{thr} then the walk goes left, otherwise it goes right. It is clear by Lemma 16 that random variable $\log_d N^{t_i}$ dominates Z^i. Hence, the probability that the population survives is at least the probability that Z^i never reaches zero ($Z^i > 0$ for all $i \in \mathbb{N}$). By Lemma 17 this is at most $(\frac{pT_{thr}}{1-pT_{thr}})^{\lfloor \log_d N^0 \rfloor}$ and thus the probability of survival is at least $1 - \left(\frac{pT_{thr}}{1-pT_{thr}}\right)^{c \ln N^0}$ where $c = \left(\frac{n\tau W_{\min}}{\ln(2n)}\right)$ depends on n, τ and fitness matrices W^e (the minimum $W_{\min} = \min_e W_{\min}^e$, and also from Lemma 16 we have that $\ln d \approx \frac{\ln 2n}{W_{\min} n\tau}$). ◀

7 Convergence of Discrete Replicator Dynamics with Mutation in Fixed Environments

In this section we extend the convergence result of Mehta et al. [22] for dynamics (1) in static environment to dynamics governed by (12) where mutations are also present. The former result critically hinges on the fact that mean fitness strictly increases unless the system is at a fixed-point, and thereby acts as a potential function. Despite the fact that this is no longer the case when mutations are introduced, we manage to show that the system still converges and follows an intuitively clear behavior. Namely, in every step of the dynamic, either the average fitness $\mathbf{x}^\top W\mathbf{y}$ or the product of the proportions of all different alleles $\prod_i x_i \prod_i y_i$ (or both) will increase. This latter quantity is, in some sense, a measure of how mixed/diverse the population is. To argue this we apply the following inequality due to Baum and Eagon:

▶ **Theorem 19** (Baum and Eagon Inequality [4]). *Let $P(\mathbf{x}) = P(\{x_{ij}\})$ be a polynomial with nonnegative coefficients homogeneous of degree d in its variables $\{x_{ij}\}$. Let $\mathbf{x} = \{x_{ij}\}$ be any point of the domain $D : x_{ij} \geq 0, \sum_{j=1}^{q_i} x_{ij} = 1, i = l, \ldots, p, j = l, \ldots, q_i$. For $\mathbf{x} = \{x_{ij}\} \in D$, let $\Xi(\mathbf{x}) = \Xi\{x_{ij}\}$ denote the point of D whose i, j coordinate is*

$$\Xi(\mathbf{x})_{ij} = \left(x_{ij} \frac{\partial P}{\partial x_{ij}}\bigg|_{(\mathbf{x})} \right) \cdot \left(\sum_{j=1}^{q_i} x_{ij} \frac{\partial P}{\partial x_{ij}}\bigg|_{(\mathbf{x})} \right)^{-1}.$$

Then $P(\Xi(\mathbf{x})) > P(\mathbf{x})$ unless $\Xi(\mathbf{x}) = \mathbf{x}$.

We will establish a potential function P that for the dynamics governed by (12), capturing sexual evolution with mutation. This will imply convergence for the dynamics. Note that feasible values of τ are in $[0, \frac{1}{n}]$, since τ represents fraction of allele i mutating to allele i' of the same gene implying $n * \tau \leq 1$.

▶ **Theorem 20** (Main 3). *Given a static environment W, dynamics governed by (12) with mutation parameter $\tau \leq \frac{1}{n}$ has a potential function $P(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^\top W\mathbf{y})^{1-n\tau} \prod_i x_i^\tau \prod_i y_i^\tau$ that strictly increases unless an equilibrium (fixed-point) is reached. Thus, the system converges to equilibria, in the limit. Equilibria are exactly the set of points $(\mathbf{p}^*, \mathbf{q}^*)$ that satisfy for all $i, i' \in S_1$, $j, j' \in S_2$:*

$$\frac{(W\mathbf{q}^*)_i}{1 - \frac{\tau}{p_i^*}} = \frac{(W\mathbf{q}^*)_{i'}}{1 - \frac{\tau}{p_{i'}^*}} = \frac{\mathbf{p}^{*T} W\mathbf{q}^*}{1 - n\tau} = \frac{(W^\top\mathbf{p}^*)_j}{1 - \frac{\tau}{q_j^*}} = \frac{(W^\top\mathbf{p}^*)_{j'}}{1 - \frac{\tau}{q_{j'}^*}}.$$

As a consequence of the above theorem we get the following:

▶ **Corollary 21.** *Along every nontrivial trajectory of dynamics governed by (12) at least one of average population fitness $\mathbf{x}^\top W\mathbf{y}$ or product of allele frequencies $\prod_i x_i \prod_i y_i$ strictly increases at each step.*

8 Conclusion and Open problems

In this paper we study various aspects of discrete replicator-like/MWUA dynamics and show three results: Two for dynamics with fixed parameters, and one where the parameters evolve over time as per a Markov chain. Theorem 12 establishes that a noisy version of discrete replicator dynamics converges *polynomially fast* to pure fixed points in coordination games. Due to the connections established by Chastain et al. [8], this implies that evolution

under sexual reproduction in haploids converges fast to a monomorphic population if the environment is static (fitness/payoff matrix is fixed). Introducing mutations to this model, as in [13], augments the replicator dynamics, and our second result shows convergence for this augmented replicator in coordination games. The proof is via a novel potential function, which is a combination of mean payoff and entropy, which may be of independent interest.

Finally, for the replicator dynamics with noise, capturing finite populations, we show that assuming some conditions (see Section B for discussion on the assumptions), the population size will eventually become zero with probability one (extinction) under (standard) replicator, while under augmented replicator (with mutations) it will never whither out (survival) with a non-trivial probability.

A host of novel questions arise from this model and there is much space for future work.

- For the fast convergence result (first result above), we assumed that the random noise δ lies in a subset of hypercube of length δ, i.e., every entry δ_i is ± 1 times magnitude δ and $\sum_i \delta_i = 0$. Can the result be generalized for a different class of random noise, where the noise also depends on the distribution of the alleles at every step and or population size?
- The second result talks about convergence to fixed points, which happens at the limit (time $t \to \infty$). Therefore, an interesting question would be to settle the speed of convergence. Additionally, for the *no mutations* case the result of [22] shows that all the stable fixed points are pure. It would be interesting to perform stability analysis for the *replicator with mutations* as well.
- Mutation can be modeled in an alternative way, where an individual can mutate to a completely new allele that is not part of some in advance fixed set of alleles. It will be interesting to define and analyze dynamics where the number of different alleles in nature is not known a priori. Finally, what happens if environment changes are not completely independent but are instead affected by population size?

Acknowledgements. We are grateful to Yuri Lyubich for helpful discussions. This work was completed while Ioannis Panageas was a PhD student at Georgia Institute of Technology. Ioannis Panageas would like to acknowledge NSF EAGER award grants CCF-1415496, CCF-1415498 and a MIT-SUTD postdoctoral fellowship. Georgios Piliouras would like to acknowledge SUTD grant SRG ESD 2015 097 and MOE AcRF Tier 2 Grant 2016-T2-1-170. Part of the work was completed while Ruta Mehta, Ioannis Panageas and Georgios Piliouras were visiting scientists at the Simons Institute for the Theory of Computing. Part of the work was completed while Ruta Mehta and Georgios Piliouras were visiting scientists at the Hausdorff Research Institute for Mathematics (HIM) during the Trimester Program on Combinatorial Optimization. Vijay Vazirani was supported by NSF Grant CCF-1216019.

References

1 H. Ackermann, P. Berenbrink, S. Fischer, and M. Hoefer. Concurrent imitation dynamics in congestion games. *Proceedings of the 28th ACM symposium on Principles of distributed computing (PODC)*, pages 63–72, 2009.

2 S. Arora, M. Hardt, and N. Vishnoi. Off the convex path, 2015. URL: `http://www.offconvex.org`.

3 S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta algorithm and applications. Technical report, Princeton, 2005.

4 L. Baum and J. Eagon. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bull. Amer. Math. Soc.*, 73:360–363, 1967.

5 J. Branke and W. Wang. *Genetic and Evolutionary Computation — GECCO 2003: Genetic and Evolutionary Computation Conference Chicago, IL, USA, July 12–16, 2003 Proceedings, Part I*, chapter Theoretical Analysis of Simple Evolution Strategies in Quickly Changing Environments, pages 537–548. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

6 M. Braverman, A. Grigo, and C. Rojas. Noise vs computational intractability in dynamics. *Innovations in Theoretical Computer Science Conference (ITCS)*, pages 128–141, 2012.

7 E. Chastain, A. Livnat, C. H. Papadimitriou, and U. V. Vazirani. Multiplicative updates in coordination games and the theory of evolution. *Innovations in Theoretical Computer Science Conference (ITCS)*, pages 57–58, 2013.

8 E. Chastain, A. Livnat, C.H. Papadimitriou, and U. Vazirani. Algorithms, games, and evolution. *Proceedings of the National Academy of Sciences (PNAS)*, 2014.

9 S. Chien and A. Sinclair. Convergence to approximate nash equilibria in congestion games. *Games and Economic Behavior*, pages 169–178, 2007.

10 Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

11 Willliam Feller. *An introduction to probability theory and its applications*, volume 2. John Wiley & Sons, 2008.

12 R. Ge, F. Huang, C. Jin, and Y. Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. *Conference on Learning Theory (COLT)*, 2015.

13 J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, Cambridge, 1998.

14 R. Kleinberg, G. Piliouras, and É. Tardos. Multiplicative updates outperform generic no-regret learning in congestion games. *ACM Symposium on Theory of Computing (STOC)*, 2009.

15 J. D. Lee, M. Simchowitz, M. I Jordan, and B. Recht. Gradient descent converges to minimizers. *Conference on Learning Theory (COLT)*, 2016.

16 A. ML Liekens. *Evolution of finite populations in dynamic environments*. Technische Universiteit Eindhoven, 2005.

17 A. Livnat, C. H. Papadimitriou, J. Dushoff, and M. W. Feldman. A mixability theory for the role of sex in evolution. *Proceedings of the National Academy of Sciences (PNAS)*, 105(50):19803–19808, 2008.

18 A. Livnat, C.H. Papadimitriou, A. Rubinstein, A. Wan, and G. Valiant. Satisfiability and evolution. *IEEE Symposium on. Foundations of Computer Science (FOCS)*, 2014.

19 V. Losert and E. Akin. Dynamics of games and genes: Discrete versus continuous time. *Journal of Mathematical Biology*, 1983.

20 Y. Lyubich. *Mathematical Structures in Population Genetics*. Springer-Verlag, 1992.

21 Y. Lyubich, G.Maistrovski, and Yu.Ol'khovski. Problems of information transmission, 1980.

22 R. Mehta, I. Panageas, and G. Piliouras. Natural selection as an inhibitor of genetic diversity: Multiplicative weights updates algorithm and a conjecture of haploid genetics. *Innovations in Theoretical Computer Science (ITCS)*, 2015.

23 R. Mehta, I. Panageas, G. Piliouras, and S. Yazdanbod. The Computational Complexity of Genetic Diversity. *European Symposium on Algorithms (ESA)*, 2016. `arXiv:1411.6322`.

24 R. Meir and D. Parkes. A note on sex, evolution, and the multiplicative updates algorithm. *Proceedings of the 12th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2015.

25 James Meiss. *Differential Dynamical Systems*. SIAM, 2007.

26 T Nagylaki. The evolution of multilocus systems under weak selection. *Genetics*, 134(2):627–47, 1993.

27 M. A. Nowak, N. L. Komarova, and P. Niyogi. Evolution of universal grammar. *Science*, 2001.

28 I. Panageas and G. Piliouras. Average Case Performance of Replicator Dynamics in Potential Games via Computing Regions of Attraction. *ACM Conference on Economics and Computation (EC)*, 2016.

29 I. Panageas and G. Piliouras. Gradient descent only converges to minimizers: Non-isolated critical points and invariant regions. *Innovations in Theoretical Computer Science (ITCS)*, 2017.

30 I. Panageas, P. Srivastava, and N. K. Vishnoi. Evolutionary dynamics in finite populations mix rapidly. *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 480–497, 2016.

31 I. Panageas and N. K. Vishnoi. Mixing time of markov chains, dynamical systems and evolution. *International Colloquium on Automata, Languages and Programming (ICALP)*, 2016.

32 R. Pemantle. Nonconvergence to unstable points in urn models and stochastic approximations. *The Annals of Probability*, 18(2), 1990.

33 O. Rivoire and S. Leibler2. The value of information for populations in varying environments. *ArXiv e-prints*, 2010.

34 L. G. Valiant. Evolvability. *J. ACM*, 56(1), 2009.

35 D. M. Wolf, V. V. Vazirani, and A. P. Arkin. Diversity in times of adversity: probabilistic strategies in microbial survival games. *Journal of theoretical biology*, 234(2):227–253, 2005.

36 S. Yang, Y. Ong, and Y. Jin. *Evolutionary computation in dynamic and uncertain environments*, volume 51. Springer Science & Business Media, 2007.

A Terms Used From Biology

We provide brief non-technical definitions of a few biological terms useful for this paper.

Gene. A unit that determines some characteristic of the organism, and passes traits to offsprings. All organisms have genes corresponding to various biological traits, some of which are instantly visible, such as eye color or number of limbs, and some of which are not, such as blood type.

Allele. Allele is one of a number of alternative forms of the same gene, found at the same place on a chromosome, Different alleles can result in different observable traits, such as different pigmentation.

Genotype. The genetic constitution of an individual organism.

Phenotype. The set of observable characteristics of an individual resulting from the interaction of its genotype with the environment.

Diploid. Diploid means having two copies of each chromosome. Almost all of the cells in the human body are diploid.

Haploid. A cell or nucleus having a single set of unpaired chromosomes. Our sex cells (sperm and eggs) are haploid cells that are produced by meiosis. When sex cells unite during fertilization, the haploid cells become a diploid cell.

B Discussion on the Assumptions and Examples

In this section, we will discuss why our assumptions are necessary and their significance.

B.1 On the parameters $\gamma, \delta, \beta, \tau$

The effective range of δ is $o\left(\frac{1}{n}\right)$, where $||\delta||_\infty = \delta$, whereas for γ is $O\left(\frac{1}{n^2}\right)$. For example, if we consider the entries of fitness matrices W^e to be uniform from interval $(1-\sigma, 1+\sigma)$ for some positive $\sigma > 0$ then γ is of $\Theta(\frac{1}{n^2})$ order. If the entries of the matrix are constants (in weak selection scenario they lie in the interval $(1-\sigma, 1+\sigma)$) then the convergence time of dynamics 3 is polynomial w.r.t n (size of fitness matrices W^e is $n \times n$). We note that the main result of [22] for dynamics (1) has been derived under the assumption that the entries of the fitness matrix are all distinct. It is proven that this assumption is necessary by giving examples where the dynamic doesn't converge to pure fixed points if the fitness matrix has some entries that are equal (the trivial example is when W has all entries equal, then every frequency vector in Δ is a fixed point). This is an indication that γ is needed to analyse the running time and is not artificial. The noise vector δ has coordinates $\pm\delta$, so it is uniformly chosen from hypercube, but there is no dependence on the current frequency vector (δ is independent of current $(\mathbf{x}, \mathbf{y})$). Finally, β should be thought of as a small constant number (like in weak selection) independent of n, and τ to be $O(\frac{1}{n})$ ($1 - n\tau \geq 0$ must hold so that the dynamics with mutation are meaningful and from Lemma 16, it must hold that $\tau \leq \frac{\beta}{16n}$).

B.2 On the environments

We analyze a finite population model where N^t is the population size at time t. It is natural to define survival if $N^t \geq 1$ for all $t \in \mathbb{N}$ (number of people is at least 1 at all times) and extinction if $N^t < 1$ for some t (if the number of people is less than one at some point then the population goes extinct). As described in preliminaries, $N^t = N^{t-1} \cdot \Phi^t$ where $\Phi^t = x(t)^\top W^{e(t)} y(t)$ is the average fitness at time t and $W^{e(t)}$ is the fitness matrix of environment $e(t)$ (at time t).

Fix a fitness matrix W (i.e., fix an environment). If $W_{ij} > 1 + \epsilon$ for all (i, j) then $\mathbf{x}^\top W \mathbf{y} \geq 1$ for all $(\mathbf{x}, \mathbf{y}) \in \Delta$ and thus the number of individuals is increasing along the generations by a factor of $1 + \epsilon$ (the population survives). On the other hand, if $W_{ij} < 1 - \epsilon$ for all (i, j) then $\mathbf{x}^\top W \mathbf{y} < 1 - \epsilon$ for all $(\mathbf{x}, \mathbf{y}) \in \Delta$, so it is clear that the number of individuals is decreasing with a factor of $1 - \epsilon$ (thus population goes extinct). So either extreme makes the problem irrelevant.

Finally, it is natural to assume that complete diversity should favor survival, i.e., if the population is uniform along the alleles/types then the population size must not decrease in the next generation. Therefore, we assume that the average fitness under uniform frequencies is $\geq 1 + \beta$ (for all but few number of bad alleles that can be seen as deleterious). The alleles that are good should dominate entry-wise the bad alleles. Example Figure 2 in the appendix C shows that this assumption is necessary. In Figure 2, $\tau = 0.03$ and $W^e = \begin{pmatrix} 0.99 & 0.37 \\ 0.56 & 2.09 \end{pmatrix}$. If we start from any vector $(\mathbf{x}, \mathbf{y})$ in the shaded area, the dynamics converges to the stable fixed point B. The average fitness $\mathbf{x}^\top W \mathbf{y}$ at B is less than the maximum at the corner which is $W^e_{1,1} = 0.99 < 1$. So if the size of population is Q when entering e, after t generations on the environment e, the population size will be at most $Q \cdot 0.99^t$ (which decreases exponentially). In that case Theorem 18 doesn't hold, even though $\frac{0.99+0.37+0.56+2.09}{4} = 1.0025 > 1$ and $\beta = 0.0025$ (qualitatively we would have the same picture for any $\tau \in [0, 0.03]$ and W^e).

The assumption defined in (7) is necessary as well for the following reason: Assume there is a combination of alleles (i,j) so that $\prod_e (W^e_{ij})^{\pi_e} \geq 1$ (*). In that case we can have one of the environments so that $x_i = 1, y_j = 1$ is a stable fixed point and hence there are initial frequencies so that the dynamics (3) converge to it. After that, it is easy to argue that this monomorphic population survives on average because of (*), so the probability of survival in that case is non zero.

B.3 Explanation of Figure 1

Figure 1 shows the adjacency graph of a Markov chain. There are 3 environments with fitness matrices, say $W^{e_1}, W^{e_2}, W^{e_3}$, and the entries of every matrix are distinct. Take $p_{ii} = 1-p$ and $p_{ij} = \frac{p}{2}$ so that the stationary distribution is $(1/3, 1/3, 1/3)$. Observe that $W^{e_1}_{1,1} \cdot W^{e_2}_{1,1} \cdot W^{e_3}_{1,1} = 1.12 \cdot 1.02 \cdot 0.87 < 0.994 < 1$. The same is true for entries (1,2),(2,1),(2,2). So the assumption defined in (7) is satisfied.

Moreover, observe that if we choose $\beta = 0.005$ and hence $\tau = \frac{0.005}{32}$ it follows that the assumptions defined in (8) are satisfied (also the bad alleles are dominated entry-wise by the good alleles). Hence, in case of no mutation, from theorem 14 the population dies out with probability 1 for all initial population sizes N^0 and all initial frequency vectors in Δ. In case of mutation, and for sufficiently large initial population size N^0, for all initial frequency vectors in Δ the probability of survival is positive (Theorem 18).

C Missing proofs

C.1 Proof of Lemma 4

Proof. From the definition of g (equation 1) we get,

$$
\begin{aligned}
&2\left(\hat{\mathbf{x}}^\top W \hat{\mathbf{y}}\right)\left(\mathbf{x}^\top W \mathbf{y}\right)^2 = 2\sum_{ij} W_{ij}\hat{x}_i\hat{y}_j\left(\mathbf{x}^\top W\mathbf{y}\right)^2 \\
&= 2\sum_{ij} W_{ij}x_iy_j \frac{(W\mathbf{y})_i}{\mathbf{x}^\top W\mathbf{y}} \frac{(W^\top\mathbf{x})_j}{\mathbf{x}^\top W\mathbf{y}}\left(\mathbf{x}^\top W\mathbf{y}\right)^2 \\
&= 2\sum_{i,j} W_{ij}x_iy_j(W\mathbf{y})_i(W^\top\mathbf{x})_j \\
&= \sum_{i,j,k} W_{ij}W_{ik}x_iy_jy_k(W^\top\mathbf{x})_j + \sum_{i,j,k} W_{ij}W^\top_{jk}x_ix_ky_j(W\mathbf{y})_i \\
&= \sum_{i,j,k} W_{ij}W_{ik}x_iy_jy_k\frac{1}{2}((W^\top\mathbf{x})_j + (W^\top\mathbf{x})_k) + \sum_{i,j,k} W_{ij}W_{kj}x_ix_ky_j\frac{1}{2}((W\mathbf{y})_i + (W\mathbf{y})_k) \\
&\geq \sum_{i,j,k} W_{ij}W_{ik}x_iy_jy_k\sqrt{(W^\top\mathbf{x})_j(W^\top\mathbf{x})_k} + \sum_{i,j,k} W_{ij}W_{kj}x_ix_ky_j\sqrt{(W\mathbf{y})_i(W\mathbf{y})_k} \\
&= \sum_i x_i\left(\sum_j y_jW_{ij}\sqrt{(W^\top\mathbf{x})_j}\right)^2 + \sum_j y_j\left(\sum_i x_iW_{ij}\sqrt{(W\mathbf{y})_i}\right)^2 \\
&\geq \left(\sum_{i,j} x_iy_jW_{ij}\sqrt{(W^\top\mathbf{x})_j}\right)^2 + \left(\sum_{j,i} y_jx_iW_{ij}\sqrt{(W\mathbf{y})_i}\right)^2 \quad \text{from convexity of } f(z)=z^2 \\
&= \left(\sum_j y_j(W^\top\mathbf{x})_j^{3/2}\right)^2 + \left(\sum_i x_i(W\mathbf{y})_i^{3/2}\right)^2. \qquad (0)
\end{aligned}
$$

Let ξ be a random variable that takes value $(W\mathbf{y})_i$ with probability x_i. Then $\mathbb{E}[\xi] = \mathbf{x}^\top W\mathbf{y}$, $\mathbb{V}[\xi] = \sum_i x_i((W\mathbf{y})_i - \mathbf{x}^\top W\mathbf{y})^2$ and ξ takes values in the interval $[0,\mu]$ with $\mu = \max_{ij} W_{ij}$. Consider the function $f(z) = z^{3/2}$ on the interval $[0,\mu]$ and observe that $f''(z) \geq \frac{3}{4}\frac{1}{\sqrt{\mu}}$ on $[0,\mu]$ since $\mu \geq \mathbf{p}^\top W\mathbf{q} \geq 0$ for all $(\mathbf{p},\mathbf{q}) \in \Delta$. Observe also that $f(\mathbb{E}[\xi]) = (\mathbf{x}^\top W\mathbf{y})^{3/2}$ and $\mathbb{E}[f(\xi)] = \sum_i x_i (W\mathbf{y})_i^{3/2}$.

▶ **Claim 22.** *$\mathbb{E}[f(\xi)] \geq f(\mathbb{E}[\xi]) + \frac{A}{2}\mathbb{V}[\xi]$, where $A = \frac{3}{4\sqrt{\mu}}$.*

Proof. By Taylor expansion we get that (we expand w.r.t $\mathbb{E}[\xi]$)

$$f(z) \geq f(\mathbb{E}[\xi]) + f'(\mathbb{E}[\xi])(z - \mathbb{E}[\xi]) + \frac{A}{2}(z - \mathbb{E}[\xi])^2$$

and hence we have that:

$$\begin{aligned} f(z) \geq f(\mathbb{E}[\xi]) + f'(\mathbb{E}[\xi])(z - \mathbb{E}[\xi]) + \frac{A}{2}(z - \mathbb{E}[\xi])^2 &\overset{\text{taking expectation}}{\Rightarrow} \\ \mathbb{E}[f(\xi)] \geq \mathbb{E}[f(\mathbb{E}[\xi])] + f'(\mathbb{E}[\xi])(\mathbb{E}[\xi] - \mathbb{E}[\xi]) + \frac{A}{2}\mathbb{V}[\xi] & \\ = f(\mathbb{E}[\xi]) + \frac{A}{2}\mathbb{V}[\xi]. & \end{aligned}$$

◀

Using the above claim it follows that:

$$\sum_i x_i (W\mathbf{y})_i^{3/2} \geq (\mathbf{x}^\top W\mathbf{y})^{3/2} + \frac{3}{8\sqrt{\mu}} \sum_i x_i ((W\mathbf{y})_i - \mathbf{x}^\top W\mathbf{y})^2.$$

Squaring both sides and omitting one square from the r.h.s we get

$$\left(\sum_i x_i (W\mathbf{y})_i^{3/2}\right)^2 \geq (\mathbf{x}^\top W\mathbf{y})^3 + \frac{3}{4\sqrt{\mu}} (\mathbf{x}^\top W\mathbf{y})^{3/2} \sum_i x_i ((W\mathbf{y})_i - \mathbf{x}^\top W\mathbf{y})^2. \tag{9}$$

We do the same by setting ξ to be $(W^\top\mathbf{x})_i$ with probability y_i and using similar argument we get

$$\left(\sum_i y_i (W^\top\mathbf{x})_i^{3/2}\right)^2 \geq (\mathbf{x}^\top W\mathbf{y})^3 + \frac{3}{4\sqrt{\mu}} (\mathbf{x}^\top W\mathbf{y})^{3/2} \sum_i y_i ((W^\top\mathbf{x})_i - \mathbf{x}^\top W\mathbf{y})^2. \tag{10}$$

Therefore it follows that

$$\begin{aligned} 2(\hat{\mathbf{x}}^\top W\hat{\mathbf{y}})(\mathbf{x}^\top W\mathbf{y})^2 &\geq \left(\sum_j y_j (W^\top\mathbf{x})_j^{3/2}\right)^2 + \left(\sum_i x_i (W\mathbf{y})_i^{3/2}\right)^2 \text{ by inequality (0)} \\ &\overset{(9)+(10)}{\geq} 2(\mathbf{x}^\top W\mathbf{y})^3 + \frac{3}{4\sqrt{\mu}} (\mathbf{x}^\top W\mathbf{y})^{3/2} \Big(\sum_i x_i \left((W\mathbf{y})_i - \mathbf{x}^\top W\mathbf{y}\right)^2 \\ &\quad + \sum_i y_i \left((W^\top\mathbf{x})_i - \mathbf{x}^\top W\mathbf{y}\right)^2\Big) \end{aligned}$$

Finally we devide both sides by $2(\mathbf{x}^\top W\mathbf{y})^2$ and we get that

$$\begin{aligned}(\hat{\mathbf{x}}^\top W\hat{\mathbf{y}}) &\geq (\mathbf{x}^\top W\mathbf{y}) + \frac{3}{8\sqrt{\mu(\mathbf{x}^\top W\mathbf{y})}}\Big(\sum_i x_i\left((W\mathbf{y})_i - \mathbf{x}^\top W\mathbf{y}\right)^2 \\ &\quad + \sum_i y_i\left((W^\top\mathbf{x})_i - \mathbf{x}^\top W\mathbf{y}\right)^2\Big) \\ &\geq (\mathbf{x}^\top W\mathbf{y}) + \frac{3}{8\mu}\left(\sum_i x_i\left((W\mathbf{y})_i - \mathbf{x}^\top W\mathbf{y}\right)^2 + \sum_i y_i\left((W^\top\mathbf{x})_i - \mathbf{x}^\top W\mathbf{y}\right)^2\right)\end{aligned}$$

with $\frac{3}{8\sqrt{\mu\Phi(\mathbf{x},\mathbf{y})}} \geq \frac{3}{8\mu}$ since $\mu \geq \mathbf{x}^\top W\mathbf{y}$. This inequality and the proof techniques can be seen as a generalization of an inequality and proof techniques in [20]. ◀

C.2 Proof of Lemma 7

Proof. Vectors $(\delta_\mathbf{x}, \delta_\mathbf{y}), (-\delta_\mathbf{x}, \delta_\mathbf{y}), (\delta_\mathbf{x}, -\delta_\mathbf{y}), (-\delta_\mathbf{x}, -\delta_\mathbf{y})$ appear with the same probability, and observe that

$$\begin{aligned}&(\mathbf{x}+\delta_\mathbf{x})^\top W(\mathbf{y}+\delta_\mathbf{y}) + (\mathbf{x}-\delta_\mathbf{x})^\top W(\mathbf{y}+\delta_\mathbf{y}) + (\mathbf{x}+\delta_\mathbf{x})^\top W(\mathbf{y}-\delta_\mathbf{y}) \\ &+ (\mathbf{x}-\delta_\mathbf{x})^\top W(\mathbf{y}-\delta_\mathbf{y}) = 4\mathbf{x}^\top W\mathbf{y},\end{aligned}$$

and the claim follows. ◀

C.3 Proof of Lemma 9

Proof. Assume w.l.o.g that we have $W_{i1} \geq W_{i2} \geq ...$ (otherwise we permute them so that are in decreasing order). Consider the case where the signs are revealed one at a time, in the order of indices of the sorted row. The probability that $+$ signs dominate $-$ signs through the process is $\frac{1}{m/2+1}$ (ballot theorem/Catalan numbers) (see 8). It is clear that when the $+$ signs dominate the $-$ signs then

$$(W\delta_\mathbf{y})_i = \sum_j^m W_{ij}\delta_j \quad \geq \quad \sum_{j=1}^{m/2}(W_{i(2j-1)} - W_{i(2j)})\delta \quad \geq \quad \gamma\delta\frac{m}{2}.$$ ◀

C.4 Proof of Lemma 11

Proof. First of all, since the average fitness is increasing in every generation (before adding noise) and by Lemma 7 we get that for all $t \in \{0, ..., 2T\}$

$$\mathbb{E}[\Phi^{t+1}|\Phi^t] \geq \Phi^t$$

namely the average fitness is a submartingale (0).

Let $(\mathbf{x}^t, \mathbf{y}^t) := (\mathbf{x}(t), \mathbf{y}(t))$ be the frequency vector at time t which has average fitness $\Phi^t \equiv \Phi(\mathbf{x}^t, \mathbf{y}^t) = \mathbf{x}^{t\top} W\mathbf{y}^t$ (abusing notation we use $\Phi(\mathbf{x}, \mathbf{y})$ for function $\mathbf{x}^\top W\mathbf{y}$ and Φ^t for the value of average fitness at time t), also we denote $(\hat{\mathbf{x}}^t, \hat{\mathbf{y}}^t) = g(\mathbf{x}^t, \mathbf{y}^t)$ and recall that $(\mathbf{x}^{t+1}, \mathbf{y}^{t+1}) = (\hat{\mathbf{x}}^t + \delta^t_\mathbf{x}, \hat{\mathbf{y}}^t + \delta^t_\mathbf{y})$. Assume that in the next generation $(\hat{\mathbf{x}}^{2t}, \hat{\mathbf{y}}^{2t}) = g(\mathbf{x}^{2t}, \mathbf{y}^{2t})$ the average fitness before the noise, namely $\hat{\mathbf{x}}^{2t\,T} W\hat{\mathbf{y}}^{2t}$ will be at least $\Phi^{2t} + C\delta\alpha^2$. Hence by

Lemma 7 we get that $\mathbb{E}[\Phi^{2t+1}|\Phi^{2t}] = \hat{\mathbf{x}}^{2t\,T} W \hat{\mathbf{y}}^{2t} \geq \Phi^{2t} + C\delta\alpha^2$ (1). Therefore we have that

$$\begin{aligned}
\mathbb{E}[\Phi^{2t+2}|\Phi^{2t}] &= \mathbb{E}_{\delta^{2t+1},\delta^{2t}}[(\hat{\mathbf{x}}^{2t+1} + \delta_{\mathbf{x}}^{2t+1})^\top W(\hat{\mathbf{y}}^{2t+1} + \delta_{\mathbf{y}}^{2t+1})|\Phi^{2t}] \\
&= \mathbb{E}_{\delta^{2t}}[\left(\hat{\mathbf{x}}^{2t+1}\right)^\top W \hat{\mathbf{y}}^{2t+1}|\Phi^{2t}] \\
&\geq \mathbb{E}_{\delta^{2t}}[\left(\mathbf{x}^{2t+1}\right)^\top W \mathbf{y}^{2t+1}|\Phi^{2t}] \\
&= \mathbb{E}[\Phi^{2t+1}|\Phi^{2t}] \\
&\geq \Phi^{2t} + C\delta\alpha^2
\end{aligned}$$

where the second inequality is claim (1) and the first inequality comes from inequality 4 (since the r.h.s of inequality 4 is non-negative). The first, third equality comes from model definition and second equality comes from Lemma 7.

Assume now that in the next generation $(\hat{\mathbf{x}}^{2t}, \hat{\mathbf{y}}^{2t}) = g(\mathbf{x}^{2t}, \mathbf{y}^{2t})$ the average fitness before the noise, namely $\hat{\mathbf{x}}^{2t\,T} W \hat{\mathbf{y}}^{2t}$ will be less than $\Phi^{2t} + C\delta\alpha^2$. This means that the vector $(\mathbf{x}^{2t}, \mathbf{y}^{2t})$ is α-close by corollary 6, so after adding the noise by the definition of α-close we get that $\hat{\mathbf{x}}^{2t\,T} W \hat{\mathbf{y}}^{2t} + \alpha \geq \Phi^{2t+1} \geq \hat{\mathbf{x}}^{2t\,T} W \hat{\mathbf{y}}^{2t} - \alpha$ (2). From Lemma 9 we will have with probability at least $\frac{1}{2}\frac{1}{m/2+1}$ that $(W\mathbf{y}^{2t+1})_i \geq (W\hat{\mathbf{y}}^{2t})_i + \frac{\gamma\delta m}{2}$ for all i in the support of vector $\mathbf{x}^t$ (we multiplied the probability by $\frac{1}{2}$ since you perturb $\mathbf{y}$ with probability half) (3). The same argument works if we purturb $\mathbf{x}$, so w.l.o.g we work with purturbed vector $\mathbf{y}$ which has support of size at least 2. Essentially by inequality 4 we get the following inequalities:

$$\begin{aligned}
\mathbb{E}[\Phi^{2t+2}|\Phi^{2t}] &= \mathbb{E}_{\delta^{2t+1},\delta^{2t}}[(\hat{\mathbf{x}}^{2t+1} + \delta_{\mathbf{x}}^{2t+1})^\top W(\hat{\mathbf{y}}^{2t+1} + \delta_{\mathbf{y}}^{2t+1})|\Phi^{2t}] \\
&= \mathbb{E}_{\delta^{2t}}[\left(\hat{\mathbf{x}}^{2t+1}\right)^\top W \hat{\mathbf{y}}^{2t+1}|\Phi^{2t}] \\
&\overbrace{\geq}^{4} \mathbb{E}_{\delta^{2t}}[\left(\mathbf{x}^{2t+1}\right)^\top W \mathbf{y}^{2t+1}|\Phi^{2t}] + \\
&+ C \cdot \mathbb{E}_{\delta^{2t}}\left[\sum_i x_i^{2t+1} \cdot \left((W\mathbf{y}^{2t+1})_i - \left(\mathbf{x}^{2t+1}\right)^\top W \mathbf{y}^{2t+1}\right)^2 \middle| \Phi^{2t}\right] \\
&\geq \Phi^{2t} + \frac{C}{m+2}\left(\frac{\gamma\delta m}{2} - 2\alpha\right)^2 \\
&\geq \Phi^{2t} + C\delta\alpha^2
\end{aligned}$$

where last inequality comes from the assumption and the second comes from claim (0), (2), (3). Hence by induction we get that

$$\mathbb{E}[\Phi^{2t+2} - (t+1)\cdot C\delta\alpha^2|\Phi^{2t}] \geq \Phi^{2t} - t \cdot C\delta\alpha^2.$$

It is easy to see that $W_{\max} \geq \Phi^t \geq W_{\min}$ for all t. ◀

C.5 Proof of Lemma 15

Proof. Consider one step of the dynamics that starts at $(\mathbf{x}, \mathbf{y})$ and has frequency vector $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ in the next step before adding the noise. Let $i*$ be the bad allele that has the greatest

fitness at it, namely $(W^e\mathbf{y})_{i*} \geq (W^e\mathbf{y})_i$ for all $i \in B_1^e$. It holds that

$$\begin{aligned}
\sum_{i\in B_1^e} \tilde{x}_i &= (1-n\tau)\sum_{i\in B_1^e} x_i \frac{(W^e\mathbf{y})_i}{\mathbf{x}^\top W^e\mathbf{y}} + \tau|B_1^e| \\
&= (1-n\tau)\frac{\sum_{i\in B_1^e} x_i(W^e\mathbf{y})_i}{\sum_{i\in G_1\setminus B_1^e} x_i(W^e\mathbf{y})_i + \sum_{i\in B_1^e} x_i(W^e\mathbf{y})_i} + \tau|B_1^e| \\
&\leq (1-n\tau)\frac{\sum_{i\in B_1^e} x_i(W^e\mathbf{y})_{i*}}{\sum_{i\in G_1\setminus B_1^e} x_i(W^e\mathbf{y})_i + \sum_{i\in B_1^e} x_i(W^e\mathbf{y})_{i*}} + \tau|B_1^e| \qquad (*) \\
&\leq (1-n\tau)\frac{\sum_{i\in B_1^e} x_i(W^e\mathbf{y})_{i*}}{\sum_{i\in G_1\setminus B_1^e} x_i(W^e\mathbf{y})_{i*} + \sum_{i\in B_1^e} x_i(W^e\mathbf{y})_{i*}} + \tau|B_1^e| \\
&= (1-n\tau)\frac{(W^e\mathbf{y})_{i*}\sum_{i\in B_1^e} x_i}{(W^e\mathbf{y})_{i*}\sum_i x_i} + \tau|B_1^e| \\
&= (1-n\tau)\sum_{i\in B_1^e} x_i + \tau|B_1^e|
\end{aligned}$$

where inequality (*) is true because if $\frac{a}{b} < 1$ then $\frac{a}{b} < \frac{a+c}{b+c}$ for all a, b, c positive. Hence after we add noise δ with $||\delta||_\infty = \delta$, the resulting vector $(\mathbf{x}', \mathbf{y}')$ (which is the next generation frequency vector) will satisfy $\sum_{i\in B_1^e} x_i' \leq (1-n\tau)\sum_{i\in B_1^e} x_i + \tau|B_1^e| + \delta|B_1^e|$. By setting $S_t = \sum_{i\in B_1^e} x_i(t)$ it follows that $S_{t+1} \leq (1-n\tau)S_t + (\tau+\delta)|B_1^e|$ and also $S_0 \leq 1$. Therefore $S_t \leq (\tau+\delta)|B_1^e|\frac{1-(1-n\tau)^t}{n\tau} + (1-n\tau)^t$. By choosing $t = -\frac{\ln(2n)}{\ln(1-n\tau)} \approx \frac{\ln(2n)}{n\tau}$ it follows that $\sum_{i\in B_1^e} x_i(t) \leq \frac{(1+o(1))|B_1^e|+1/2}{n} \leq \frac{2|B_1^e|}{n}$ where we used the assumption that $\delta = o_n(\tau)$. The same argument holds for B_2^e. ◀

C.6 Proof of Lemma 16

Proof. By Lemma 15 after $\frac{\ln(2n)}{n\tau}$ generations it follows that

$$\sum_{i\in B_1^e} x_i(t) + \sum_{j\in B_2^e} y_i(t) \leq \frac{2|B^e|}{n}. \tag{11}$$

We consider the average fitness function $\mathbf{x}^\top W^e\mathbf{y}$ which is not increasing (as has already been mentioned). Let $\tau = \tau\cdot(1,...,1)^\top$, $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) = f(\mathbf{x}, \mathbf{y})$ and $(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = g(\mathbf{x}, \mathbf{y})$ with fitness matrix W^e and also denote by $(\mathbf{x}', \mathbf{y}')$ the resulting vector after noise δ is added. It is easy to observe that

$$\tilde{\mathbf{x}}^\top W^e\tilde{\mathbf{y}} = (1-n\tau)^2\hat{\mathbf{x}}^\top W^e\hat{\mathbf{y}} + (1-n\tau)\hat{\mathbf{x}}^\top W^e\tau + (1-n\tau)\tau^\top W^e\hat{\mathbf{y}} + \tau^\top W^e\tau,$$

and also that

$$\mathbf{x}'^\top W^e\mathbf{y}' \geq \tilde{x}^\top W^e\tilde{y} - 2n\delta W^e_{\max} \geq \tilde{x}^\top W^e\tilde{y}\left(1 - O\left(2n\delta\frac{W_{max}}{W_{min}}\right)\right) = (1-o_{n\tau}(1))\tilde{x}^\top W\tilde{y},$$

where $W_{\max} = \max_e W^e_{\max}$. Under the assumption 8 we have the following upper bounds:

- $\hat{\mathbf{x}}^\top W^e\tau \geq (1+\beta)n\tau\left(1 - \frac{2|B_1^e|}{n}\right)$ and $\hat{\tau}^\top W^e\hat{\mathbf{y}} \geq (1+\beta)n\tau\left(1 - \frac{2|B_2^e|}{n}\right)$.
- $\tau^\top W^e\tau \geq (n\tau)^2(1+\beta)\left(1 - \frac{|B^e|}{n}\right) \geq (1+\beta)\left(1 - \frac{2|B^e|}{n}\right)^2 n^2\tau^2$.

First assume that $\mathbf{x}^\top W^e \mathbf{y} \leq 1 + \frac{\beta}{2}$. We get the following system of inequalities:

$$\begin{aligned}
\frac{\mathbf{x}'^\top W^e \mathbf{y}'}{\mathbf{x}^\top W^e \mathbf{y}} &\geq (1 - o_{n\tau}(1)) \frac{\tilde{\mathbf{x}}^\top W^e \tilde{\mathbf{y}}}{\mathbf{x}^\top W^e \mathbf{y}} \\
&\geq (1 - o_{n\tau}(1))((1-n\tau)^2 \frac{\hat{\mathbf{x}}^\top W^e \hat{\mathbf{y}}}{\mathbf{x}^\top W^e \mathbf{y}} + 2(1-n\tau)n\tau \left(1 - \frac{2|B^e|}{n}\right) \frac{(1+\beta)}{\mathbf{x}^\top W^e \mathbf{y}} + \\
&\qquad + \frac{(1+\beta)}{\mathbf{x}^\top W^e \mathbf{y}} \left(1 - \frac{2|B^e|}{n}\right)^2 n^2\tau^2) \\
&\geq (1 - o_{n\tau}(1))((1-n\tau)^2 + 2(1-n\tau)n\tau \left(1 - \frac{2|B^e|}{n}\right)\left(1 + \frac{\beta}{2+\beta}\right) + \\
&\qquad + \left(1 + \frac{\beta}{2+\beta}\right)\left(1 - \frac{2|B^e|}{n}\right)^2 n^2\tau^2) \\
&\geq (1 - o_{n\tau}(1)) \left(1 + n\tau \left(\frac{2\beta}{2+\beta} - \frac{6|B^e|}{n} - \frac{2\beta}{2+\beta} n\tau\right)\right) \\
&\geq 1 + n\tau \left(\frac{\beta}{2+\beta}\right)
\end{aligned}$$

The second inequality comes from the fact that $\hat{\mathbf{x}}^\top W^e \hat{\mathbf{y}} \geq \mathbf{x}^\top W^e \mathbf{y}$ (the average fitness is increasing for the no mutation setting) and also since $\mathbf{x}^\top W^e \mathbf{y} \leq 1 + \frac{\beta}{2}$. The third and the fourth inequality use the fact that $|B^e| \ll n\beta$ and $\tau \leq \frac{\beta}{16n}$. Therefore, the fitness increases in the next generation for the mutation setting as long as the current fitness $\mathbf{x}^\top W^e \mathbf{y} \leq 1 + \frac{\beta}{2}$ with a factor of $1 + n\tau\frac{\beta}{2+\beta}$ (i). Hence the time we need to reach the value of 1 for the average fitness is $\frac{2\ln\frac{1}{h}}{n\tau\frac{\beta}{2+\beta}}$ which is dominated by $t_1 = \frac{\ln(2n)}{n\tau}$. Therefore the total loss factor is at most $\frac{1}{d} = h^{t_1}$, namely $d = \left(\frac{1}{h}\right)^{t_1}$. Let t_2 be the time for the average fitness to reach $1 + \frac{\beta}{4}$ (as long as it has already reached 1), thus $t_2 = \frac{2}{n\tau}$ which is dominated by t_1. By similar argument, let's now assume that $\mathbf{x}^\top W^e \mathbf{y} \geq 1 + \frac{\beta}{2}$ then

$$\begin{aligned}
\frac{\mathbf{x}'^\top W^e \mathbf{y}'}{\mathbf{x}^\top W^e \mathbf{y}} &\geq (1 - o_{n\tau}(1)) \left(\frac{\tilde{\mathbf{x}}^\top W^e \tilde{\mathbf{y}}}{\mathbf{x}^\top W^e \mathbf{y}}\right) \\
&\geq (1 - o_{n\tau}(1))((1-n\tau)^2 \frac{\hat{\mathbf{x}}^\top W^e \hat{\mathbf{y}}}{\mathbf{x}^\top W^e \mathbf{y}} + 2(1-n\tau)n\tau \left(1 - \frac{2|B^e|}{n}\right) \frac{(1+\beta)}{\mathbf{x}^\top W^e \mathbf{y}} \\
&+ \frac{(1+\beta)}{\mathbf{x}^\top W^e \mathbf{y}} \left(1 - \frac{2|B^e|}{n}\right)^2 n^2\tau^2) \\
&\geq 1 - 2n\tau
\end{aligned}$$

Hence $\mathbf{x}'^\top W^e \mathbf{y}' \geq (1 - 2n\tau)(1 + \frac{\beta}{2})$, namely $\mathbf{x}'^\top W^e \mathbf{y}' \geq 1 + \frac{\beta}{4}$ (ii) for $\tau < \frac{\beta}{16n}$. Therefore as long as the fitness surpasses $1 + \frac{\beta}{4}$, it never goes below $1 + \frac{\beta}{4}$ (conditioned on the fact you remain at the same environment). This is true from claims (i), (ii). When the fitness is at most $1 + \frac{\beta}{2}$, it increases in the next generation and when it is greater than $1 + \frac{\beta}{2}$, it remains at least $1 + \frac{\beta}{4}$ in the next generation. To finish the proof we compute the times. The time t_3 to have a total gain factor of at least d, will be such that $(1 + \frac{\beta}{4})^{t_3} = \frac{1}{h^{t_1}}$. Hence $t_3 = t_1 \frac{2\ln\frac{1}{h}}{\beta}$. By setting $T_{thr} = \frac{6\ln(2n)}{n\tau\beta W_{\min}} > \frac{6\ln\frac{1}{h}\ln(2n)}{n\tau\beta} > 3t_3 > t_1 + t_2 + t_3$ the proof finishes. ◀

C.7 Proof of Theorem 20

Proof. We first prove the results for rational τ; let $\tau = \frac{\kappa}{\lambda}$. We use the theorem of Baum and Eagon [4]. Let

$$L(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^\top W \mathbf{y})^{\lambda - m\kappa} \prod_i x_i^\kappa \prod_i y_i^\kappa.$$

Then

$$x_i \frac{\partial L}{\partial x_i} = 2\kappa L + \frac{2x_i (W\mathbf{y})_i (\lambda - m\kappa) L}{\mathbf{x}^\top W \mathbf{y}}.$$

It follows that

$$\begin{aligned} \frac{x_i \frac{\partial L}{\partial x_i}}{\sum_i x_i \frac{\partial L}{\partial x_i}} &= \frac{2\kappa L + \frac{2x_i (W\mathbf{y})_i (\lambda - m\kappa) L}{\mathbf{x}^\top W \mathbf{y}}}{2m\kappa L + 2(\lambda - m\kappa) L} \\ &= \frac{2\kappa L}{2\lambda L} + \frac{2L(\lambda - m\kappa) x_i (W\mathbf{y})_i}{2\lambda L \mathbf{x}^\top W \mathbf{y}} \\ &= (1 - n\tau) x_i \frac{(W\mathbf{y})_i}{\mathbf{x}^\top W \mathbf{y}} + \tau \end{aligned}$$

where the first equality comes from the fact that $\sum_{i=1}^n x_i (W\mathbf{y})_i = \mathbf{x}^\top W \mathbf{y}$. The same is true for $y_i \frac{\partial L}{\partial y_i}$. Since L is a homogeneous polynomial of degree 2λ, from Theorem 19 we get that L is strictly increasing along the trajectories, namely

$$L(f(\mathbf{x}, \mathbf{y})) > L(\mathbf{x}, \mathbf{y})$$

unless $(\mathbf{x}, \mathbf{y})$ is a fixed point (f is the update rule of the dynamics, see also 2). So $P(\mathbf{x}, \mathbf{y}) = L^{\frac{1}{\kappa}}(\mathbf{x}, \mathbf{y})$ is a potential function for the dynamics.

To prove the result for irrational τ, we just have to see that the proof of [4] holds for all homogeneous polynomials with degree d, even irrational.

To finish the proof let $\Omega \subset \Delta$ be the set of limit points of an orbit $\mathbf{z}(t) = (\mathbf{x}(t), \mathbf{y}(t))$ (frequencies at time t for $t \in \mathbb{N}$). $P(\mathbf{z}(t))$ is increasing with respect to time t by above and so, because P is bounded on Δ, $P(\mathbf{z}(t))$ converges as $t \to \infty$ to $P^* = \sup_t \{P(\mathbf{z}(t))\}$. By continuity of P we get that $P(\mathbf{v}) = \lim_{t\to\infty} P(\mathbf{z}(t)) = P^*$ for all $\mathbf{v} \in \Omega$. So P is constant on Ω. Also $\mathbf{v}(t) = \lim_{k\to\infty} \mathbf{z}(t_k + t)$ as $k \to \infty$ for some sequence of times $\{t_i\}$ and so $\mathbf{v}(t)$ lies in Ω, i.e. Ω is invariant. Thus, if $\mathbf{v} \equiv \mathbf{v}(0) \in \Omega$ the orbit $\mathbf{v}(t)$ lies in Ω and so $P(\mathbf{v}(t)) = P^*$ on the orbit. But P is strictly increasing except on equilibrium orbits and so Ω consists entirely of fixed points. ◀

C.8 Calculations for mutation

Let $(\hat{\mathbf{x}}, \hat{\mathbf{y}}) = g(\mathbf{x}, \mathbf{y})$. If in every generation allele $i \in S_1$ mutates to allele $k \in S_1$ with probability μ_{ik}, where $\sum_k \mu_{ik} = 1$, $\forall i$, then the final proportion (after reproduction, mutation) of allele $i \in S_1$ in the population will be

$$x_i' = \sum_{k \in S_1} \mu_{ki} \hat{x}_k.$$

Similarly, if $j \in S_2$ mutates to $k \in S_2$ with probability δ_{jk}, then proportion of allele $j \in S_2$ will be

$$y_j' = \sum_{k \in S_2} \delta_{ki} \hat{y}_k.$$

If mutation happens after every selection (mating), then we get the following dynamics with update rule $f' : \Delta \to \Delta$ governing the evolution (update rule contains selection+mutation).

$$\text{Let } (\mathbf{x}', \mathbf{y}') = f'(\mathbf{x}, \mathbf{y}), \text{ then } \begin{array}{rcl} x'_i & = & \sum_{k \in S_1} \mu_{ki} x_k \frac{(W\mathbf{y})_k}{\mathbf{x}^\top W \mathbf{y}}, \ \forall i \in S_1 \\ y'_j & = & \sum_{k \in S_2} \delta_{kj} y_k \frac{(\mathbf{x}^\top W)_k}{\mathbf{x}^\top W \mathbf{y}}, \ \forall j \leq S_2 \end{array} \tag{12}$$

Suppose $\forall k,\ \forall i \neq k$ and $\forall j \neq k$, we have $\mu_{ik} = \delta_{jk} = \tau$, where $\tau \leq \frac{1}{n}$. Since $\sum_k \mu_{ik} = \sum_k \delta_{jk} = 1$, we have $\mu_{ii} = \delta_{jj} = 1 - (n-1)\tau = 1 + \tau - n\tau$. Hence

$$\begin{aligned} x'_i &= \sum_{k \in S_1} \mu_{ki} x_k \frac{(W\mathbf{y})_k}{\mathbf{x}^\top W \mathbf{y}} \\ &= (1 + \tau - n\tau) x_i \frac{(W\mathbf{y})_i}{\mathbf{x}^\top W \mathbf{y}} + \tau \sum_{k \neq i} x_k \frac{(W\mathbf{y})_k}{\mathbf{x}^\top W \mathbf{y}} \\ &= (1 - n\tau) x_i \frac{(W\mathbf{y})_i}{\mathbf{x}^\top W \mathbf{y}} + \tau \sum_{k} x_k \frac{(W\mathbf{y})_k}{\mathbf{x}^\top W \mathbf{y}} \\ &= (1 - n\tau) x_i \frac{(W\mathbf{y})_i}{\mathbf{x}^\top W \mathbf{y}} + \tau. \end{aligned}$$

The same is true for vector $\mathbf{y}'$. The dynamics of (12) where $\mu_{ik} = \delta_{ik} = \tau$ for all $k \neq i$ simplifies to the equations 2 as appear in the preliminaries.

D Figures

To draw the phase portrait of a discrete time system $f : \Delta \to \Delta$, we draw vector $f(\mathbf{x}) - \mathbf{x}$ at point $\mathbf{x}$.

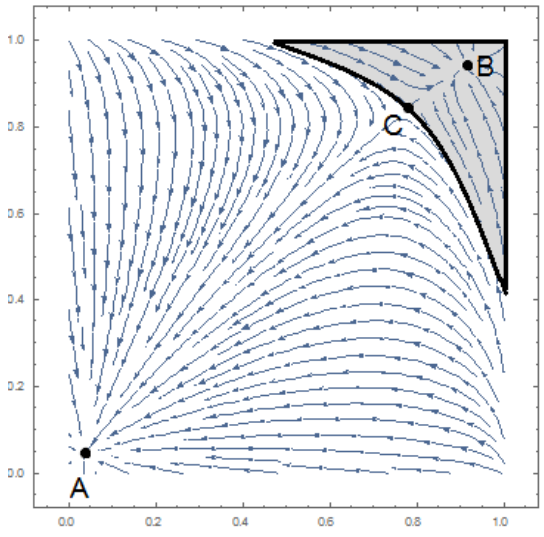


Figure 2 Example where population goes extinct in environment e for some initial frequency vectors $(\mathbf{x}, \mathbf{y})$ that are close to stable point B (inside the shaded area). Mutation probability is $\tau = 0.03$ and the fitness matrix of environment e is $W^e_{1,1} = 0.99, W^e_{2,2} = 2.09, W^e_{1,2} = 0.37, W^e_{2,1} = 0.56$.

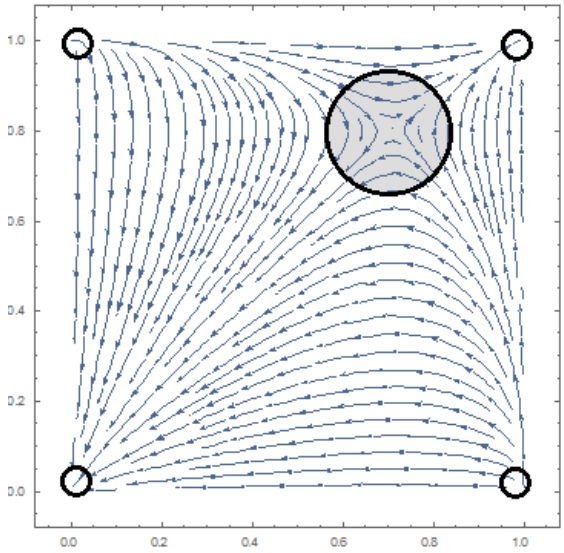


Figure 3 Example of dynamics without mutation in specific environment $W^e_{1,1} = 0.99, W^e_{2,2} = 2.09, W^e_{1,2} = 0.37, W^e_{2,1} = 0.56$. The circles qualitatively show all the points that slow down the increase in the average fitness $\mathbf{x}^T W^e \mathbf{y}$, i.e α-close points or negligible.

Self-Sustaining Iterated Learning*

Bernard Chazelle[1] and Chu Wang[2]

1 **Princeton University, Princeton, USA**
chazelle@cs.princeton.edu
2 **Nokia Bell Labs, Murray Hill, USA**
chu.wang@nokia-bell-labs.com

Abstract

An important result from psycholinguistics (Griffiths & Kalish, 2005) states that no language can be learned iteratively by rational agents in a self-sustaining manner. We show how to modify the learning process slightly in order to achieve self-sustainability. Our work is in two parts. First, we characterize iterated learnability in geometric terms and show how a slight, steady increase in the lengths of the training sessions ensures self-sustainability for any discrete language class. In the second part, we tackle the nondiscrete case and investigate self-sustainability for iterated linear regression. We discuss the implications of our findings to issues of non-equilibrium dynamics in natural algorithms.

1998 ACM Subject Classification I.2.6 Learning

Keywords and phrases Iterated learning, language evolution, iterated Bayesian linear regression, non-equilibrium dynamics

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.17

1 Introduction

Consider this hypothetical scenario: A native speaker of Quenya[1] sets out to teach the language to an English speaker; after a year of teaching, the learner considers herself fluent enough to teach Quenya to some other English speaker, who a year later does the same. In this form of *iterated learning*, agents teach each other in sequence: X teaches Y, who then teaches Z, who then teaches...[2, 8, 7, 15, 12, 9, 14, 16, 18, 11]. By a classic result of Griffiths and Kalish [7], Quenya will vanish after a finite number of iterations, at which point the agents, assumed to be rational, will be "teaching" each other plain English. In other words, after a while, learners will be taught nothing they don't already know: iterated learning is not self-sustaining.

Such findings are hard to validate empirically but variants of it are within the reach of experimental psychology. As early as 1932, in fact, the English psychologist Frederic Bartlett used iterated learning to expose hidden biases among humans. He presented a picture of an owl to a person for given period of time and then asked her to draw it from memory. Her picture was then shown to the next learner for the same amount of time, who then proceeded to draw it back from memory. After 20 iterations of this process, to Bartlett's surprise, what was being drawn was no longer an owl but, quite clearly, a cat! The challenge was to explain why humans would exhibit a pro-feline bias without falling into the trap of just-so stories.

* This work was supported in part by NSF grant CCF-1420112.
[1] Quenya is one of J.R.R. Tolkien's fictional languages.


© Bernard Chazelle and Chu Wang;
licensed under Creative Commons License CC-BY

Griffiths et al. [11] repeated the same experiment ten years ago, this time trading owls for lines. The goal was to see if linear regression could be iterated: the answer was a resounding No. Skipping over logistical details, the experiment presents the first learner with a cloud of 20 points drawn randomly, with noise, from the line $Y = 1 - X$. The cloud vanishes and the learner is then asked to reconstruct it from memory. She then becomes the teacher by passing on her own cloud to the next learner, who likewise, looks at it for a while, and then tries to reconstruct it from memory, etc. Surprisingly, iterating this process a mere nine times leads the last learner in the sequence to draw a cloud that regresses to the line $Y = X$; in other words, teaching about descending lines iteratively has precisely the opposite effect! In fact the initial picture is essentially is irrelevant. A random cloud of points will also lead to $Y = X$.

Unlike the Quenya scenario, where the bias toward English is not unexpected, the cat and line experiments both reveal a hidden prior among the participants. Humans seem to love cats and possess a strong positive correlation bias; it is easy to speculate why.[2] It is noteworthy that the prior should prevail even in the absence of any sort of priming. Indeed, this experiment fails miserably if you try it yourself by playing the role of all the agents in sequence. The use of different learners ensures that the training does not acquire long-term memory. Similar laboratory experiments with human subjects (well, undergraduates) have confirmed the unstainability of iterated learning [11, 2, 19, 1, 9].

In our first example, Quenya gets “washed out" by English in a way reminiscent of the fixation of an allele through genetic drift. Indeed, the original impetus for studying iterated learning in psycholinguistics was to look for a parallel to Kimura’s neutral theory of molecular evolution in the area of cultural transmission. People learn their native tongue from speakers who themselves learned it from others. This process introduces variation along the way, some of which is retained durably. The selectionist view seeks to explain this process by fitness considerations at the population level. Iterated learning suggests a different explanation. Language acquisition suffers from a well-documented information bottleneck (the notorious “poverty of stimulus"), so one might expect languages to evolve so as to be easy to learn: could complexity theory be the key? This push for simplicit would then trigger the emergence of linguistic universals (eg, compositionality) that one finds present in all languages [8]. This view complements – some will argue, contradict – Chomsky’s interpretation of universals as the product of constraints imposed by an innate genetic endowment.

Following Chomsky and Lasnik’s theory of “Principles and Parameters," Rafferty et al. [15] model languages by means of a handful of parameters: think of a few knobs whose settings specify any given language. Language evolution thus entails the trans-generational update of a probability distribution over that parameter space. Assuming that the learners are rational Bayesian agents, iterated learning acts as a Gibbs sampler for a joint probability distribution over languages and their sentences. By converging to a stationary distribution, iterated learning proves incapable of sustaining itself past the mixing time. In that model, languages evolve to reflect the priors of the learners while losing all trace of the ancestor language. While this phenomenon is of central relevance in the study of universal grammars, it leaves open the possibility that changes in the sampling algorithm might make iterated learning self-sustaining. Of course, it is easy to think of situations where this feature would be highly desirable (eg, school teaching, social transmission of norms, legends, jokes, etc.) We show how keeping the length of the training sessions growing slightly allows iterated

[2] Our favorite piece of anecdotal evidence in support of the positive slope bias is that no road sign in the US features an aircraft on a descending path.

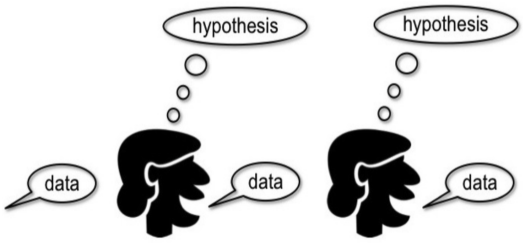


Figure 1 Chained iterated learning.

learning to be sustained in perpetuity.

In the first part of the paper, we characterize iterated learnability in geometric terms and show how a slight, steady increase in the lengths of the learning sessions ensures self-sustainability for any discrete language class. In the second part, we tackle the nondiscrete case and investigate self-sustainability for iterated Bayesian linear regression. In all cases, self-sustainability requires making the underlying Markov process time-inhomogeneous in order to stay out of equilibrium. This gives us an opportunity to offer a few thoughts on the growing importance of non-equilibrium in natural algorithms.

Background

Following [2, 8, 7, 15, 12, 9, 14, 16, 18, 11], we begin with *chained iterated learning*: a learner's prior is modeled by a distribution over a hypothesis space $\mathcal{H}$, which is itself equipped with a likelihood function: $\mathbb{P}[d|h]$ indicates the probability of generating data $d \in \mathcal{D}$ given the hypothesis $h \in \mathcal{H}$. The first learner samples m_1 items *iid* from the initial hypothesis $\boldsymbol{h}_{\texttt{init}}$: these items provide the training data $\mathbf{d}_1 = (d_{1,1}, \ldots, d_{1,m_1})$ with which the first learner Bayes-updates its prior. Its posterior is given by setting $t = 1$ in this formula:

$$\mathbb{P}[h|\mathbf{d}_t] = \mathbb{P}[\mathbf{d}_t|h]\,\mathbb{P}[h]/\mathbb{P}[\mathbf{d}_t], \quad \text{with } \mathbb{P}[\mathbf{d}_t] = \sum_{h\in\mathcal{H}} \mathbb{P}[\mathbf{d}_t|h]\,\mathbb{P}[h]. \tag{1}$$

From that point on, each successive learner updates its prior from their predecessor. For any $t > 1$, learner t receives m_t items sampled from the posterior of agent $t-1$ to form the training set $\mathbf{d}_t$. To do that, she picks a random hypothesis h from $\mathcal{H}$ with probability $\mathbb{P}[h|\mathbf{d}_{t-1}]$ (the posterior of learner $t-1$) and then samples m_t items *iid* from h to form $\mathbf{d}_t \in \mathcal{D}^{m_t}$. The posterior $\mathbb{P}[h|\mathbf{d}_t]$ is derived according to (1). Note that learner t has no direct access to the posterior of learner $t-1$ but only to data drawn from a hypothesis sampled from the posterior. Our formulation assumes a discrete space $\mathcal{H}$ but extends to continuous settings, as we show in §3.

In the case of linguistic transmission, each hypothesis $h \in \mathcal{H}$ is a "knob" whose setting is given by a number between 0 and 1, specifically the prior probability $\mathbb{P}[h]$. All learners share the same prior. Picking some h from that prior specifies a *language* (also denoted h for convenience). In this case, a language is defined as a probability distribution over $\mathcal{D}$, interpreted here as a set of *sentences*. In this way, the prior can be viewed as a mixture over $\mathcal{H}$: by abuse of terminology, we call it a *mixed* hypothesis, which we distinguish from a *pure* hypothesis of the form $h \in \mathcal{H}$ (corresponding to a single-point distribution). Access to language h is achieved by random sampling: the sentence $d \in \mathcal{D}$ is picked with probability $\mathbb{P}[d|h]$.

Iterated learning proceeds as follows. After selecting language h with probability $\mathbb{P}[h|\mathbf{d}_{t-1}]$, learner t collects m_t independent samples from h. Thus, given a tuple $\mathbf{d}_t = (d_1, \ldots, d_{m_t})$ of sentences from $\mathcal{D}$, the likelihood $\mathbb{P}[\mathbf{d}_t|h]$ is equal to $\prod_{1\le k\le m_t} \mathbb{P}[d_k|h]$. The learner is now

ready to Bayes-update its prior. Of course, the first one ($t = 1$) samples directly from the language $\boldsymbol{h}_{\texttt{init}}$ chosen for iterated learning. The notation is boldfaced to indicate that $\boldsymbol{h}_{\texttt{init}}$ may be a mixed hypothesis or, in other words, a distribution over hypotheses.

Suppose that $\mathcal{D} = \{d_1, \ldots, d_s\}$ and $\mathcal{H} = \{h_1, \ldots, h_n\}$ are both finite. While sampling from the posterior of learner $t-1$, if learner t winds up choosing h_i then, by Bayesian updating, the probability $P_{ij}^{|t}$ that its posterior picks h_j is given by:

$$P_{ij}^{|t} = \sum_{\mathbf{d} \in \mathcal{D}^{m_t}} \mathbb{P}[h_j|\mathbf{d}]\, \mathbb{P}[\mathbf{d}|h_i] = \sum_{\mathbf{d} \in \mathcal{D}^{m_t}} \frac{\mathbb{P}[\mathbf{d}|h_i]\, \mathbb{P}[\mathbf{d}|h_j]\, \mathbb{P}[h_j]}{\sum_{k=1}^{n} \mathbb{P}[\mathbf{d}|h_k]\, \mathbb{P}[h_k]}\,. \tag{2}$$

To our knowledge, the entire literature on the topic assumes a common, fixed sample size for all the learners: $m_t = m$. Equation (2) can be then interpreted as marginalizing a Gibbs sampler over the data space, which creates a Markov chain over the hypothesis space $\mathcal{H}$: if $\boldsymbol{h}^t$ denotes the row vector formed by the n probabilities $\mathbb{P}[h_k \,|\, \mathbf{d}_t]$, then $\boldsymbol{h}^t = \boldsymbol{h}^{t-1} P^t$, where $\boldsymbol{h}^0 = \boldsymbol{h}_{\texttt{init}}$. Assuming ergodicity (in this case, a fairly inconsequential technical assumption), the chain can be shown to converge to a unique stationary distribution $\boldsymbol{h}$. It can be easily checked that it coincides with the prior: $\boldsymbol{h} = (\mathbb{P}[h_1], \ldots, \mathbb{P}[h_n])$ [7, 13]; see [15, 16] for an analysis of the mixing time in specific linguistic scenarios. This convergence reveals the long-term unsustainability of iterated learning. We show how diversifying the sample sizes m_t, hence making the Markov chain time-inhomogeneous, can overcome this weakness.

Our results

In §2, we show how to achieve self-sustainability in the discrete setting [8, 7], using only a logarithmically increasing sample size; specifically, the new hypothesis to be learned is acquired by all the (infinitely many) learners with probability at least $1 - \varepsilon$ using a sample size of $O(\log \frac{t}{\varepsilon})$ for the t-th learner. The constant factor depends on the geometry of the hypothesis space. By relaxing the objective and allowing learners to settle on an arbitrarily close approximation of the hypothesis to be learned, we can remove all dependency on the geometry of the hypothesis space.

In §3, we extend the iterated learning model to a Gaussian setting for an infinite hypothesis space and show that a sample size of $O(t)^{1+o(1)}$ is sufficient to ensure self-sustainability. We also show that allowing learners to pick their teachers at random cuts down the sample size to $O(\log t)^{1+o(1)}$. The arguments used for the discrete case bump into singularities so we use a different approach, which allows us to exploit various "stability" properties of the Gaussian setting.

In §4, we turn our attention to the iterated version of Bayesian linear regression and prove a high-probability statement about self-sustainability. This requires spectral arguments from random matrix theory and, in particular, bounds on the lowest singular value of Wishart matrices.

Discussion

Before moving to the technical part of this work, we add a few thoughts about its larger context and relevance. For a dynamicist, the loss of Quenya is a byproduct of the memory-erasing ergodicity implied by mixing. For a physicist, the loss is due to the Second Law of thermodynamics and the bounded supply of free energy available to each agent: together these two constraints make it impossible to keep the system out of equilibrium. For a biologist, this entropic pull toward equilibrium is the hallmark of a dying system. Evolution is nature's attempt to optimize the absorption of free energy into work while maximizing

the production of entropy. The first requirement is keeping the system out of equilibrium over timescales well in excess of the metabolic rate (here, the teaching rate). From that perspective, our work can be seen as an effort to find out the minimum conditions necessary to keep a target dynamics active in perpetuity. There are several approaches to this question and the two we follow are among the simplest: (i) increasing the supply of free energy (eg, lengthening the training sessions) and (ii) mixing timescales (eg, rewiring the communication network).

Most of the work on Markov chains in theoretical computer science regards mixing as a blessing: large spectral gaps are good while small ones are to be avoided. In biology, however, mixing often means death. In fact, much of life can be seen as nature's attempt to keep mixing at bay. This paper explores what can be done to prevent a Markov chain from reaching equilibrium. We expect this theme to gain prominence in future work on natural algorithms.

2 Self-Sustainability

We show how to make iterated learning self-sustaining in the presence of a finite hypothesis space $\mathcal{H} = \{h_1, \ldots, h_n\}$. This involves specifying a sequence of training session lengths $m_1, m_2, \ldots$ so that the posterior of any learner ends up differing from $\boldsymbol{h}_{\texttt{init}}$ by an arbitrarily small amount. Formally, given any $\delta, \varepsilon \geq 0$, we say that iterated learning is (δ, ε)-*self-sustaining* if, with probability at least $1 - \varepsilon$, a random $h \in \mathcal{H}$ picked from any learner's posterior distribution differs from $\boldsymbol{h}_{\texttt{init}}$ in total variation by at most δ. We recall a few facts: the hypothesis h denotes a language modeled as a probability distribution over $\mathcal{D}$; the total variation distance is half the ℓ_1-norm; and the posterior of learner t after the t-th iteration is defined by marginalizing $\mathbb{P}[h|\mathbf{d}_t]$ over all samples $\mathbf{d}_t$ drawn from a random h picked from the posterior of learner $t-1$ (or $\boldsymbol{h}_{\texttt{init}}$ if $t = 1$). As a shorthand, we speak of ε-self-sustainability to refer to the case $\delta = 0$.

The parameters δ and ε allow us to distinguish between two metrics: the distance between two languages over $\mathcal{D}$ and the distance between two mixtures over $\mathcal{H}$. The two notions could differ widely. For example, if all of $\mathcal{H}$ corresponds to languages very close to $\boldsymbol{h}_{\texttt{init}}$, to achieve (δ, ε)-self-sustainability might be easy for a tiny $\delta > 0$ but hopelessly difficult for $\delta = 0$. The complexity of iterated learning depends on the geometry of the languages formed by the pure hypotheses. This is best captured by introducing a metric that, though more specialized than the total variation (it works only on the simplex of probability vectors) brings all sorts of technical benefits: the *root-sine distance* between two probability distributions $\boldsymbol{a} = (a_1, \ldots, a_s)$ and $\boldsymbol{b} = (b_1, \ldots, b_s)$ over $\mathcal{D}$ is defined as

$$d_{RS}(\boldsymbol{a}, \boldsymbol{b}) = \sqrt{\frac{1}{2} \sum_{i,j=1}^{s} \left(\sqrt{a_i b_j} - \sqrt{a_j b_i} \right)^2} = \sqrt{1 - \left(\sum_{i=1}^{s} \sqrt{a_i b_i} \right)^2}. \tag{3}$$

It would be surprising if this distance had not been used before, but we could not find a reference. We prove that it is indeed a metric in the Appendix and also explain its name. We show that it is related to the Hellinger, Bhattacharyya and total variation distances, d_H, d_B, d_{TV} by the following relations:

$$\begin{cases} d_H = \sqrt{1 - \sqrt{1 - d_{RS}^2}}\,; \\ d_B = -\frac{1}{2} \ln(1 - d_{RS}^2)\,; \\ d_{TV} \leq \sqrt{2s}\, d_{RS}. \end{cases} \tag{4}$$

2.1 The results

We focus on the "pure" case $\boldsymbol{h}_{\texttt{init}} \in \mathcal{H}$, and later briefly discuss how to generalize the method to mixed hypotheses. Using the shorthand $\mathtt{d}_{ij}$ for $d_{RS}(\mathbb{P}[\cdot|h_i], \mathbb{P}[\cdot|h_j])$, we define $\mathtt{d}_i := \min_{j:j\neq i} \mathtt{d}_{ij}$. Let $\boldsymbol{p} = (p_1, p_2, \ldots, p_n)$ be the prior distribution over $\mathcal{H}$, where $p_i := \mathbb{P}[h_i]$. We can obviously assume that each p_i is positive and that all the pure hypotheses are distinct, hence $\mathtt{d}_i > 0$. The two theorems below assume that $\boldsymbol{h}_{\texttt{init}} = h_1$.

▶ **Theorem 1.**. *For any positive $\varepsilon < 1$, the following sample size sequence makes iterated learning ε-self-sustaining:*

$$m_t = \frac{4}{\mathtt{d}_1^2} \ln \frac{nt}{\varepsilon\, p_1} = \frac{4}{\mathtt{d}_1^2}\Big(\log \frac{t}{\varepsilon} + C\Big),$$

for some $C > 0$ independent of $t, \varepsilon, \mathtt{d}_1$.

The factor 4 can be reduced to $2^{1+o(1)}$ if we adjust the constant C. It is to be expected that the lengths of the training sessions should grow to infinity as p_1 tends to zero, as the vanishing prior makes it increasingly difficult for the posteriors to "attach" to h_1. The session lengths are sensitive to the minimum distance between the languages specified by $\mathcal{H}$ and the target language h_1. Settling for (δ, ε)-self-sustainability allows us to remove this dependency.

▶ **Theorem 2.**. *For any positive $\delta, \varepsilon < 1$, the following sample size sequence makes iterated learning (δ, ε)-self-sustaining:*

$$m_t = \frac{8sn^2}{\delta^2}\Big(\ln \frac{t}{\varepsilon} + C\Big).$$

for some $C > 0$ independent of t, δ, ε.

2.2 The proofs

To establish Theorem 1, we estimate the probability P^* that each leaner ends up picking h_1. Recall that $\boldsymbol{h}^t$ is the posterior distribution of learner t, by the Markovian property of the system,

$$P^* = \mathbb{P}[\boldsymbol{h}^0 = h_1] \prod_{t\geq 0} \mathbb{P}[\boldsymbol{h}^{t+1} = h_1 | \boldsymbol{h}^t = h_1] = \prod_{t\geq 1} P_{11}^{|t}. \tag{5}$$

Since the matrix $P^{|t}$ is the transition matrix of a Markov chain, we proceed by bounding its off-diagonal elements $P_{ij}^{|t}$ for $i \neq j$. By (2) and Young's inequality,

$$\begin{aligned}
P_{ij}^{|t} &\leq \sum_{\mathbf{d}\in\mathcal{D}^{m_t}} \frac{\mathbb{P}[\mathbf{d}|h_i]\,\mathbb{P}[\mathbf{d}|h_j]p_j}{\mathbb{P}[\mathbf{d}|h_i]p_i + \mathbb{P}[\mathbf{d}|h_j]p_j} = \frac{p_j}{p_i} \sum_{\mathbf{d}\in\mathcal{D}^{m_t}} \frac{\big(\frac{p_i}{p_j}\big)\mathbb{P}[\mathbf{d}|h_i]\,\mathbb{P}[\mathbf{d}|h_j]}{\big(\frac{p_i}{p_j}\big)\mathbb{P}[\mathbf{d}|h_i] + \mathbb{P}[\mathbf{d}|h_j]} \\
&\leq \frac{1}{2}\sqrt{\frac{p_j}{p_i}} \sum_{\mathbf{d}\in\mathcal{D}^{m_t}} \sqrt{\mathbb{P}[\mathbf{d}|h_i]\,\mathbb{P}[\mathbf{d}|h_j]} = \frac{1}{2}\sqrt{\frac{p_j}{p_i}}\Big(\sum_{d\in\mathcal{D}} \sqrt{\mathbb{P}[d|h_i]\,\mathbb{P}[d|h_j]}\Big)^{m_t} \\
&\leq \frac{1}{2}\sqrt{\frac{p_j}{p_i}} \exp\left\{\frac{m_t}{2}\Big(\Big(\sum_{d\in\mathcal{D}} \sqrt{\mathbb{P}[d|h_i]\,\mathbb{P}[d|h_j]}\Big)^2 - 1\Big)\right\}.
\end{aligned}$$

By definition of the root-sine distance, we have

$$P_{ij}^{|t} \leq \frac{1}{2}\sqrt{\frac{p_j}{p_i}} e^{-\frac{1}{2}\mathtt{d}_{ij}^2 m_t} \quad (i \neq j). \tag{6}$$

Setting $i = 1$ in (6) and summing over $2 \le j \le n$, it follows by Cauchy-Schwarz that

$$\sum_{j=2}^{n} P_{1j}^{|t} \le \frac{1}{2}\sqrt{\frac{n(1-p_1)}{p_1}}\, e^{-\frac{1}{2}\mathtt{d}_1^2 m_t}. \tag{7}$$

Combining (5) and (7) yields

$$P^* \ge \prod_{t\ge 1}\left(1 - \frac{1}{2}\sqrt{\frac{n(1-p_1)}{p_1}}\, e^{-\frac{1}{2}\mathtt{d}_1^2 m_t}\right) \ge 1 - \frac{1}{2}\sqrt{\frac{n(1-p_1)}{p_1}}\sum_{t\ge 1} e^{-\frac{1}{2}\mathtt{d}_1^2 m_t}. \tag{8}$$

Given $0 < \varepsilon < 1$, we constrain the sequence (m_t) to satisfy:

$$\sum_{t\ge 1} e^{-\frac{1}{2}\mathtt{d}_1^2 m_t} < \varepsilon\sqrt{\frac{4p_1}{n(1-p_1)}}. \tag{9}$$

For example, we can pick the sequence

$$m_t = \frac{1}{\mathtt{d}_1^2}\ln\frac{n(1-p_1)t^4}{\varepsilon^2 p_1},$$

which completes the proof. A closer look at the calculation shows that the factor t^4 can be reduced to $C_\alpha t^{2+\alpha}$ for any small $\alpha > 0$ and a suitable constant $C_\alpha > 0$, which makes the dependency on t arbitrarily close to $(2/\mathtt{d}_1^2)\ln t$. ◀

To prove Theorem 2, we set a target distance $\rho := \delta/(n\sqrt{2s})$ and find a subset $A \subseteq \mathcal{H}$ such that (i) $\mathtt{d}_{1j} \le \rho n$ for $j \in A$ and (ii) $\mathtt{d}_{ij} \ge \rho$ for $i \in A$ and $j \notin A$. To see why such a subset must exist, consider spheres centered at $\boldsymbol{h}_{\mathtt{init}} = h_1$ of radius $k\rho$, for $k = 1, \ldots, n+1$ (with respect to d_{RS}). These define $n+1$ disjoint (open) regions and, by the pigeonhole principle, at least one of them must be empty. We set A to include all the points in the regions preceding the empty one; note that $h_1 \in A$. The claim follows from the triangular inequality. We begin with a straightforward generalization of (7): for any $i \in A$,

$$\sum_{j\notin A} P_{ij}^{|t} \le \frac{1}{2}\sqrt{\frac{n(1-p_A)}{p_A}}\, e^{-\frac{1}{2}\rho^2 m_t}, \tag{10}$$

where $p_A := \min_{i\in A} p_i$. Now let P^* be the probability that $\boldsymbol{h}^t \in A$ for each t, then (5) and (8) are generalized to

$$P^* \ge \prod_{t\ge 1}\left(1 - \max_{i\in A}\sum_{j\notin A} P_{ij}^{|t}\right) \ge 1 - \frac{1}{2}\sqrt{\frac{n(1-p_A)}{p_A}}\sum_{t\ge 1} e^{-\frac{1}{2}\rho^2 m_t}. \tag{11}$$

Setting

$$m_t = \frac{1}{\rho^2}\ln\frac{n(1-p_A)t^4}{\varepsilon^2 p_A} \tag{12}$$

ensures that $P^* > 1-\varepsilon$. The root-sine distance between the languages denoted by h_1 and any $h \in A$ is at most ρn, so that, by (4), the total variation distance is bounded by $\sqrt{2s}\rho n = \delta$, which concludes the proof of Theorem 2. ◀

So far, we have analyzed only the "pure" case $\boldsymbol{h}_{\mathtt{init}} \in \mathcal{H}$. The idea of the training is to prevent the prior to "drag" the posterior mixture all across $\mathcal{H}$. It should be clear that a

similar result obtains if $\boldsymbol{h}_{\texttt{init}} \in \Delta\mathcal{H}$ is concentrated on a subset A of $\mathcal{H}$. The proof follows the path charted in Theorem 2 and need not be repeated here. It is crucial to note, however, that this result is to be understood in a coarse-graining sense: iterated learning cannot ensure that the original weights in the mixture $\boldsymbol{h}_{\texttt{init}}$ are retained but only that A contributes most of the mass in the posteriors. To retain the weights would require changing the stationary distribution to conform with $\boldsymbol{h}_{\texttt{init}}$, as the process unfolds, something that straightforward Bayesian learning seems unable to do. Learning pure hypotheses bypasses that difficulty.

2.3 Applications

We briefly discuss a direct application of our results to a well-known model of language acquisition via iterated learning and we mention some natural extensions of the techniques.

Language evolution

Rafferty et al. [15] show how iterated learning fails rapidly in a simple model of language evolution. Given n hypotheses, iterated learning with fixed-length training sessions ceases to learn anything new after only $O(\log n \log\log n)$ rounds. The previous theorems show how to turn this around and achieve self-sustainality. In the model, $\mathcal{H} = \{h_1, \ldots, h_n\}$, where $n = 2^k$ and h_i denotes the language whose sentences are words in $\{0, 1, ?\}^k$ with exactly m question marks and $0, 1$ matching the binary decomposition of $i-1$ outside the question marks. For example, if $k = 4$ and $m = 2$, then h_3 denotes the language

$$\{\, 00??, 0?1?, ?01?, 0??0, ?0?0, ??10 \,\}.$$

We can assume that m is much smaller than k. Each language has the same length $\binom{k}{m}$ and the total number of sentences is $s = \binom{k}{m}2^{k-m}$. The prior is given by $\mathbb{P}[h_i] = p_i = 1/n$. Given a hypothesis h_i, $\mathbb{P}[d|h_i] = 1/\binom{k}{m}$ if d has m question marks and match the bits of $i-1$ elsewhere; else it is 0 (and d, h are called incompatible). Given $h \in \mathcal{H}$,

$$\begin{cases} \mathbb{P}[d] = \sum_{h\in\mathcal{H}} \mathbb{P}[d|h]\mathbb{P}[h] = 2^{m-k}/\binom{k}{m}\,; \\ \mathbb{P}[h|d] = \mathbb{P}[d|h]\mathbb{P}[h]/\mathbb{P}[d] = 2^{-m} \quad \text{(or 0 if } d, h \text{ are incompatible).} \end{cases}$$

We easily check that $\mathtt{d}_1^2 = 1 - \left(\sum_{i=1}^s \sqrt{a_i b_i}\right)^2 \geq 1 - \left(\frac{m}{k}\right)^2 > \frac{1}{2}$; hence, by Theorem 1, session lengths m_t no larger than $O(\log \frac{t}{\varepsilon})$ are sufficient to maintain ε-self-sustainability.

Meanings and utterances

In the use of iterated learning for studying language evolution [7, 14], it is common to model the data $\mathbf{d}$ as a joint distribution $(\boldsymbol{x}, \boldsymbol{y})$ over a product space $\mathcal{X}^{m_t} \times \mathcal{Y}^{m_t}$. The idea is to distinguish between "meanings" $\boldsymbol{x}$ and "utterances" $\boldsymbol{y}$. In this setting, $\mathbb{P}[\mathbf{d}|h] = \mathbb{P}[\boldsymbol{y}|\boldsymbol{x}, h]\mu(\boldsymbol{x})$, where $\mu(\boldsymbol{x})$ is the probability of generating $\boldsymbol{x}$. The transition matrix of the Markov chain thus becomes

$$\begin{aligned} P_{ij}^{|t} &= \sum_{\boldsymbol{x}\in\mathcal{X}^{m_t}} \sum_{\boldsymbol{y}\in\mathcal{Y}^{m_t}} \mathbb{P}[h_j|\boldsymbol{x}, \boldsymbol{y}]\, \mathbb{P}[\boldsymbol{y}|\boldsymbol{x}, h_i]\mu(\boldsymbol{x}) \\ &= \sum_{\boldsymbol{x}\in\mathcal{X}^{m_t}} \sum_{\boldsymbol{y}\in\mathcal{Y}^{m_t}} \frac{\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}, h_i]\, \mathbb{P}[\boldsymbol{y}|\boldsymbol{x}, h_j]\, \mathbb{P}[h_j]}{\sum_{k=1}^m \mathbb{P}[\boldsymbol{y}|\boldsymbol{x}, h_k]\, \mathbb{P}[h_k]} \mu(\boldsymbol{x})\,. \end{aligned} \tag{13}$$

Since the output $\boldsymbol{y}$ now depends on both the hypothesis and the input data, we redefine $\mathtt{d}_{ij}$ as the root-sine distance between the two distributions $\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}, h_i]\mu(\boldsymbol{x})$ and $\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}, h_j]\mu(\boldsymbol{x})$:

$$\mathtt{d}'_{ij} := 1 - \left(\sum_{\boldsymbol{x}\in\mathcal{X}} \sum_{\boldsymbol{y}\in\mathcal{Y}} \sqrt{\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}, h_i]\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}, h_j]}\,\mu(\boldsymbol{x}) \right)^2 \tag{14}$$

and we define $\mathtt{d}'_i := \min_{j:j\neq i} \mathtt{d}'_{ij}$. Given any $i \neq j$,

$$\begin{aligned}
P_{ij}^{|t} &\leq \sum_{\boldsymbol{x}\in\mathcal{X}^{m_t}} \sum_{\boldsymbol{y}\in\mathcal{Y}^{m_t}} \frac{\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}, h_i]\,\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}, h_j]\,p_j}{\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}, h_i]\,p_i + \mathbb{P}[\boldsymbol{y}|\boldsymbol{x}, h_j]\,p_j}\mu(\boldsymbol{x}) \\
&\leq \frac{1}{2}\sqrt{\frac{p_j}{p_i}} \sum_{\boldsymbol{x}\in\mathcal{X}^{m_t}} \sum_{\boldsymbol{y}\in\mathcal{Y}^{m_t}} \sqrt{\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}, h_i]\,\mathbb{P}[\mathbf{y}|\boldsymbol{x}, h_j]}\,\mu(\boldsymbol{x}) \\
&\leq \frac{1}{2}\sqrt{\frac{p_j}{p_i}} \Big(\sum_{\boldsymbol{x}\in\mathcal{X}} \sum_{\boldsymbol{y}\in\mathcal{Y}} \sqrt{\mathbb{P}[\boldsymbol{y}|h_i]\,\mathbb{P}[\boldsymbol{y}|h_j]}\,\mu(\boldsymbol{x}) \Big)^{m_t} \\
&\leq \frac{1}{2}\sqrt{\frac{p_j}{p_i}} \exp\left\{ \frac{m_t}{2} \Big(\Big(\sum_{\boldsymbol{x}\in\mathcal{X}} \sum_{\boldsymbol{y}\in\mathcal{Y}} \sqrt{\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}, h_i]\,\mathbb{P}[\boldsymbol{y}|\boldsymbol{x}, h_j]}\,\mu(\boldsymbol{x}) \Big)^2 - 1 \Big) \right\}.
\end{aligned}$$

This gives us this new version of inequality (6), which we can use as the basis for a repeat of the argument of the previous section:

$$P_{ij}^{|t} \leq \frac{1}{2}\sqrt{\frac{p_j}{p_i}} e^{-\frac{1}{2}\mathtt{d}'^2_{ij} m_t} \qquad (i \neq j). \tag{15}$$

3 Iterated Learning in Continuous Spaces

When iterated learning operates over a hypothesis space $\mathcal{H}$ parametrized continuously, say, in $\mathbb{R}$, the minimum root-sine distance usually vanishes and the previous arguments run into singularities and collapse. A new approach is needed. To make our discussion concrete, we assume that the prior distribution of each learner is a Gaussian $\mathbb{P}[h] \sim N(\bar{\mu}, \bar{\sigma}^2)$ and that the likelihood of producing data d given hypothesis h is also normal: $\mathbb{P}[d|h] = N(h, \sigma^2)$. The likelihood can also be understood as a noisy measurement of h: $d = h + \phi$, where the noise $\phi \sim N(0, \sigma^2)$. We assume that the data received by the first learner comes from $N(\mu_0, \sigma_0^2)$. This is the simplest instance of a continuous setting in which the root-sine distance argument fails. We discuss it in some detail, considering both chained learning and its generalizations; and then we use the results to treat the case of iterated Bayesian linear regression.

During its training session, the t-th learner receives data $\mathbf{d}_t = (d_{t,1}, \ldots, d_{t,m_t})$ from its predecessor: it is obtained by first picking a random hypothesis h from the posterior of learner $t-1$ and then collecting m_t independent random samples from $N(h, \sigma^2)$. For the case $t = 1$, we can treat the original teacher as learner 0 with its posterior equal to $N(\mu_0, \sigma_0^2)$. Learner t Bayes-updates its posterior as follows:

$$\mathbb{P}[h|\mathbf{d}_t] \propto \mathbb{P}[\mathbf{d}_t|h]\mathbb{P}[h] \propto \exp\Big(-\frac{1}{2\sigma^2}\sum_{i=1}^{m_t}(d_{t,i} - h)^2\Big) \exp\Big(-\frac{1}{2\bar{\sigma}^2}(h - \bar{\mu})^2\Big),$$

which is still Gaussian, with mean and variance denoted by μ_t and σ_t^2, respectively. Carrying out the usual square completion gives up these update rules: for $t > 0$,

$$\begin{cases} \mu_t = \frac{1}{\bar{\tau}+m_t\tau}\left(\bar{\tau}\bar{\mu} + \tau(d_{t,1} + d_{t,2} + \cdots + d_{t,m_t})\right) \\ \tau_t \;= \bar{\tau} + m_t\tau, \end{cases} \tag{16}$$

where we define the precisions $\tau = 1/\sigma^2$, $\bar{\tau} = 1/\bar{\sigma}^2$, and $\tau_t = 1/\sigma_t^2$. We say that iterated learning is *ε-self-sustaining* if $|\mathbb{E}\,\mu_t - \mu_0| \le \varepsilon$ and $\sigma_t^2 + \operatorname{var}\mu_t$ remains bounded for all t. If $\sigma_t^2 + \operatorname{var}\mu_t \to 0$ as $t \to \infty$, we say that iterated learning is *strongly ε-self-sustaining*. We consider successively the case of chained iterated learning and the more challenging "hopping" scenario in which a new learner picks a random teacher from the past (instead of the previous one).

3.1 Chained learning

In chained iterated learning, the data $d_{t,i}$ is a noisy message drawn from the posterior of the $(t-1)$-th learner; hence $d_{t,i} \sim N(\mu_{t-1}, \sigma_{t-1}^2 + \sigma^2)$. In view of (16), μ_t is itself Gaussian. By taking the expectation and variance of equation (16), we find the following recursive relations for $\mathbb{E}\,\mu_t$ and $\operatorname{var}\mu_t$: for $t > 0$,

$$\begin{cases} \mathbb{E}\,\mu_t = \frac{1}{\bar{\tau}+m_t\tau}\big(\bar{\tau}\bar{\mu} + m_t\tau\,\mathbb{E}\,\mu_{t-1}\big); \\ \operatorname{var}\mu_t = \frac{m_t\tau^2}{(\bar{\tau}+m_t\tau)^2}\big(\operatorname{var}\,\mu_{t-1} + \sigma_{t-1}^2 + \sigma^2\big). \end{cases} \tag{17}$$

If we define $\beta_t := m_t\tau/(\bar{\tau} + m_t\tau)$, then (17) becomes $\mathbb{E}\,\mu_t = \beta_t\,\mathbb{E}\,\mu_{t-1} + (1-\beta_t)\bar{\mu}$. If $m_t = m$ is a constant, then so is β_t, and the recursive relation (17) becomes

$$\mathbb{E}\,\mu_t - \bar{\mu} = \beta_1^t(\mu_0 - \bar{\mu}),$$

which shows that $\mathbb{E}\,\mu_t$ converges to $\bar{\mu}$ exponentially fast. As in the discrete case, iterated learning is not self-sustainable with constant-length training sessions. By letting m_t increase as $O(t^{1+o(1)})$ order, however, we can achieve self-sustainability:

▶ **Theorem 3.** *For any $0 < \varepsilon < 1$, the following sample size sequence makes chained iterated learning strongly ε-self-sustaining:*

$$m_t = \frac{|\mu_0 - \bar{\mu}|}{\varepsilon}\Big(1 + \frac{1}{c}\Big)\Big(\frac{\sigma}{\bar{\sigma}}\Big)^2 t^{1+c},$$

for an arbitrarily small constant $c > 0$.

Proof. We observe that $\mathbb{E}\,\mu_t$ is a convex combination of $\bar{\mu}$ and $\mathbb{E}\,\mu_s$ $(s < t)$; specifically,

$$\mathbb{E}\,\mu_t = \prod_{s=1}^{t}\beta_s\mu_0 + \Big(1 - \prod_{s=1}^{t}\beta_s\Big)\bar{\mu}. \tag{18}$$

Because $\sum_{s>0}(1/s)^{1+c} < 1 + \int_1^\infty x^{-1-c}\,dx = 1 + 1/c$, we have

$$\begin{aligned} 1 \;&\ge\; \prod_{s=1}^{t}\beta_s = \prod_{s=1}^{t}\Big(1 - \frac{\bar{\tau}}{m_s\tau + \bar{\tau}}\Big) \ge 1 - \sum_{s=1}^{t}\frac{\bar{\tau}}{m_s\tau + \bar{\tau}} \\ &\ge\; 1 - \frac{\varepsilon}{|\mu_0 - \bar{\mu}|}\Big(\frac{c}{c+1}\Big)\sum_{s=1}^{\infty}\frac{1}{s^{1+c}} > 1 - \frac{\varepsilon}{|\mu_0 - \bar{\mu}|}\,. \end{aligned}$$

This shows that

$$|\mathbb{E}\,\mu_t - \mu_0| = \Big(1 - \prod_{s=1}^{t}\beta_s\Big)|\bar{\mu} - \mu_0| \le \varepsilon.$$

By (16), $\sigma_t^2 = 1/\tau_t < 1/m_t\tau \to 0$. Since $\sigma_{t-1}^2 \le \bar{\sigma}^2$ for $t > 1$, it follows from (17) that $\operatorname{var}\mu_t \le (\operatorname{var}\mu_{t-1} + \sigma^2 + \bar{\sigma}^2)/m_t$ for $t > 1$, and $\operatorname{var}\mu_1 \le (\sigma_0^2 + \sigma^2)/m_1$. Writing $M_t := m_t m_{t-1} \dots m_1$, we have

$$\begin{aligned} M_t \operatorname{var}\mu_t &\le M_{t-1}\operatorname{var}\mu_{t-1} + M_{t-1}(\sigma^2 + \bar{\sigma}^2) \\ &\le tM_{t-1}(\sigma_0^2 + \sigma^2 + \bar{\sigma}^2), \end{aligned}$$

and thus $\operatorname{var}\mu_t \le (\sigma_0^2 + \sigma^2 + \bar{\sigma}^2)t/m_t \to 0$ since $m_t = \Omega(t^{1+c})$. ◀

3.2 Hopped learning

We consider the "hopped learning" scenario in which learner t hops back to pick a teacher from $\{0, 1, \dots, t-1\}$ at random, and then samples m_t bits of data from her posterior. The recursive relation for μ_t becomes

$$\mu_t = \frac{\beta_t}{m_t} \sum_{s=0}^{t-1} \chi_{t,s} \sum_{i=1}^{m_t} d_{t,s,i} + (1 - \beta_t)\bar{\mu}, \tag{19}$$

where, given t, the random variable $\chi_{t,s}$ is 1 for a value of s picked at random between 0 and $s-1$, and is zero elsewhere; recall that $\beta_t := m_t\tau/(\bar{\tau} + m_t\tau)$. Hopped iterated learning provides access to earlier data, so one would expect the lengths of the training sessions to grow more slowly than in chained learning. The change is indeed quite dramatic:

▶ **Theorem 4.** *For any positive $\varepsilon < |\mu_0 - \bar{\mu}|$, the following sample size sequence makes hopped iterating learning ε-self-sustaining:*

$$m_t = B_c \frac{|\mu_0 - \bar{\mu}|}{\varepsilon} \left(\frac{\sigma}{\bar{\sigma}}\right)^2 (1 + \log t)^{1+c},$$

for an arbitrarily small $c > 0$ and a constant B_c that depends only on c.

Proof. By taking expectation on both sides of (19), for any $t > 0$,

$$\mathbb{E}\,\mu_t = \frac{\beta_t}{t} \sum_{s=0}^{t-1} \mathbb{E}\,\mu_s + (1 - \beta_t)\bar{\mu},$$

We define $\gamma_1 = \beta_1$ and, for $t > 1$,

$$\gamma_t := (1 + \beta_1)\left(1 + \frac{\beta_2}{2}\right) \cdots \left(1 + \frac{\beta_{t-1}}{t-1}\right) \frac{\beta_t}{t}.$$

We verify easily that $\mathbb{E}\,\mu_t = \gamma_t\mu_0 + (1 - \gamma_t)\bar{\mu}$, for $t > 0$; therefore, the first part in establishing ε-self-sustainability consists of proving that

$$1 \ge \gamma_t \ge 1 - \frac{\varepsilon}{|\mu_0 - \bar{\mu}|}, \tag{20}$$

which will show that $|\mathbb{E}\,\mu_t - \mu_0| \le \varepsilon$. Note that

$$\gamma_t \le \frac{1}{t} \prod_{s=1}^{t-1} \left(1 + \frac{1}{s}\right) = 1.$$

Now define

$$\alpha_s = \frac{\varepsilon}{B_c |\mu_0 - \bar{\mu}| s (1 + \log s)^{1+c}}.$$

for $s > 0$. We pick a constant B_c large enough so that α_s is small enough to carry out first-order Taylor approximations around $1 + \alpha_s$. We find that

$$1 + \frac{\beta_s}{s} = 1 + \frac{1}{s}\Big(1 - \frac{1}{1 + m_t\tau/\bar\tau}\Big) \ge \Big(1 + \frac{1}{s}\Big)\Big(1 - \frac{1}{(s+1)m_t\tau/\bar\tau}\Big)$$
$$\ge \Big(1 + \frac{1}{s}\Big)\Big(1 - \frac{s\alpha_s}{s+1}\Big) \ge \Big(1 + \frac{1}{s}\Big)(1 - \alpha_s) \ge \Big(1 + \frac{1}{s}\Big)e^{-2\alpha_s}.$$

Thus,

$$\gamma_t \ge \frac{\beta_t}{t}\prod_{s=1}^{t-1}\Big(1 + \frac{1}{s}\Big)e^{-2\sum_{s=1}^{t-1}\alpha_s} = \beta_t e^{-2\sum_{s=1}^{t-1}\alpha_s} \ge 1 - \frac{\varepsilon}{|\mu_0 - \bar\mu|},$$

which establishes (20). Our derivation relies on the fact that

$$\beta_t \ge 1 - \frac{\varepsilon}{B_c|\mu_0 - \bar\mu|(1 + \log t)^{1+c}} \ge 1 - \frac{\varepsilon}{2|\mu_0 - \bar\mu|}$$

and

$$\sum_{s=1}^{t-1}\frac{1}{s(1+\log s)^{1+c}} \le 1 + \frac{1}{(\log e)^{1+c}}\int_2^{t-1}\frac{1}{x(\ln x)^{1+c}}\,dx = O\Big(\frac{1}{c}\Big);$$

hence,

$$e^{-2\sum_{s=1}^{t-1}\alpha_s} \ge e^{-O(\varepsilon/(cB_c|\mu_0-\bar\mu|))} \ge 1 - \frac{\varepsilon}{2|\mu_0 - \bar\mu|}.$$

Having shown that $|\mathbb{E}\,\mu_t - \mu_0| \le \varepsilon$ for all t, it now suffices to prove that $\sigma_t^2 + \operatorname{var}\mu_t$ remains bounded. We note that $\tau_t > m_t\tau \to \infty$, hence $\sigma_t^2 = 1/\tau_t \to 0$, so the remainder of the proof needs to establish that the variance of μ_t stays bounded. Writing $D_{t,s} := d_{t,s,1} + \cdots + d_{t,s,m_t}$, we have $\operatorname{var} D_{t,s} = m_t \operatorname{var} d_{t,s,1} = m_t(\sigma_s^2 + \sigma^2 + \operatorname{var}\mu_s)$; hence

$$\mathbb{E}\,D_{t,s}^2 = \operatorname{var} D_{t,s} + (\mathbb{E}\,D_{t,s})^2 = m_t(\sigma_s^2 + \sigma^2 + \operatorname{var}\mu_s) + m_t^2(\mathbb{E}\,\mu_s)^2.$$

In (19), the variables $\chi_{t,s}$ and $D_{t,s}$ are independent, for $0 \le s \le t-1$; furthermore, $\mathbb{E}\,\chi_{t,s} = \mathbb{E}\,\chi_{t,s}^2 = 1/t$, and $\mathbb{E}\,\chi_{t,s_1}\chi_{t,s_2} = 0$ if $s_1 \ne s_2$; therefore,

$$\operatorname{var}[\chi_{t,s}D_{t,s}] = \mathbb{E}\,\chi_{t,s}^2\,\mathbb{E}\,D_{t,s}^2 - (\mathbb{E}\,\chi_{t,s})^2(\mathbb{E}\,D_{t,s})^2 = \frac{\mathbb{E}\,D_{t,s}^2}{t} - \frac{(\mathbb{E}\,D_{t,s})^2}{t^2}$$
$$= \Big(\frac{m_t}{t}\Big)\big(\sigma_s^2 + \sigma^2 + \operatorname{var}\mu_s + m_t(\mathbb{E}\,\mu_s)^2\big) - \Big(\frac{m_t}{t}\Big)^2(\mathbb{E}\,\mu_s)^2 \tag{21}$$

and, for $s_1 \ne s_2$,

$$\begin{aligned}\operatorname{cov}[\chi_{t,s_1}D_{t,s_1}, \chi_{t,s_2}D_{t,s_2}] &= \mathbb{E}[\chi_{t,s_1}\chi_{t,s_2}D_{t,s_1}, D_{t,s_2}] - \mathbb{E}[\chi_{t,s_1}D_{t,s_1}]\mathbb{E}[\chi_{t,s_2}D_{t,s_2}]\\ &= \mathbb{E}[\chi_{t,s_1}\chi_{t,s_2}]\mathbb{E}[D_{t,s_1}D_{t,s_2}] - \mathbb{E}\,\chi_{t,s_1}\mathbb{E}\,D_{t,s_1}\mathbb{E}\,\chi_{t,s_2}\mathbb{E}\,D_{t,s_2}\\ &= -\frac{1}{t^2}\mathbb{E}\,D_{t,s_1}\mathbb{E}\,D_{t,s_2} = -\Big(\frac{m_t}{t}\Big)^2\mathbb{E}\,\mu_{s_1}\mathbb{E}\,\mu_{s_2}.\end{aligned} \tag{22}$$

Then, by taking the variance on both sides of (19), we have

$$\begin{aligned}
\operatorname{var}\mu_t &= \Big(\frac{\beta_t}{m_t}\Big)^2 \operatorname{var}\sum_{s=0}^{t-1}\chi_{t,s}D_{t,s}\\
&= \Big(\frac{\beta_t}{m_t}\Big)^2\Big(\sum_{s=0}^{t-1}\operatorname{var}\left[\chi_{t,s}D_{t,s}\right] + \sum_{0\le s_1\ne s_2\le t-1}\operatorname{cov}\left[\chi_{t,s_1}D_{t,s_1},\chi_{t,s_2}D_{t,s_2}\right]\Big)\\
&= \Big(\frac{\beta_t}{m_t}\Big)^2\Big(\sum_{s=0}^{t-1}\Big(\frac{m_t}{t}\Big)\big(\sigma_s^2+\sigma^2+\operatorname{var}\mu_s+m_t(\mathbb{E}\,\mu_s)^2\big) - \Big(\frac{m_t}{t}\Big)^2\Big(\sum_{s=0}^{t-1}\mathbb{E}\,\mu_s\Big)^2\Big)\\
&\le \frac{1}{tm_t}\sum_{s=0}^{t-1}\big(\sigma_s^2+\sigma^2+\operatorname{var}\mu_s+m_t(\mathbb{E}\,\mu_s)^2\big).
\end{aligned}$$

Notice that $\sigma_s^2\to 0$ and $(\mathbb{E}\,\mu_s)^2$ is bounded since $|\mathbb{E}\,\mu_t-\mu_0|\le\varepsilon$. We conclude that $\sigma_t^2+\operatorname{var}\mu_t$ remains bounded for all t. ◀

4 Iterated Bayesian Linear Regression

The iterated version of Bayesian linear regression has been the subject of extensive study in the field of psychology [11, 2, 19, 1, 9]. The work has involved experimentation with human subjects but little in the way of theoretical analysis. This section is a first step toward filling this void. The task at hand is to estimate a hypothesis $h\in\mathcal{H}:=\mathbb{R}^d$ given a noisy measurements on the hyperplane $y=h^Tx$, where $x\in\mathbb{R}^d$. In the Bayesian setting, we assume a Gaussian prior on the hypothesis space: $\mathbb{P}[h]\sim N(\bar\mu,\bar\sigma^2 I_d)$. The data is given by (x,y), where $x\sim N(0,I_d)$ and $y=h^Tx+\phi$, for $\phi\sim N(0,\sigma^2)$ (with x,ϕ independent). Since we typically make several measurements, we write this (likelihood) relation in matrix form: $y=Xh+\phi$, where $y\in\mathbb{R}^m$ (with m the number of measurements); $\phi\sim N(0,\sigma^2 I_m)$; and X is an m-by-d matrix each of whose rows denotes a random vector $x\sim N(0,I_d)$. This means that the matrix X is random (a fact of key importance in our discussion below). We have:

$$\begin{cases}
\mathbb{P}[\phi]\sim \exp\{-\frac{1}{2\sigma^2}\|\phi\|_2^2\} & \texttt{(noise)}\\
\mathbb{P}[h]\sim \exp\{-\frac{1}{2\bar\sigma^2}\|h-\bar\mu\|_2^2\} & \texttt{(prior)}\\
\mathbb{P}[y|X,h]\sim \exp\{-\frac{1}{2\sigma^2}\|y-Xh\|_2^2\} & \texttt{(likelihood)}
\end{cases}$$

In iterated Bayesian linear regression, the t-th learner receives her data from learner $t-1$. Here, learner 0 is treated just like any other agent, except that his prior $\mathbb{P}[h]\sim N(\mu_0,\bar\sigma^2 I_d)$ is the distribution to be learned iteratively. Since sampling from the prior is independent of X, Bayesian updating gives the posterior $N(\mu_t,\Sigma_t)$, where

$$\mathbb{P}[h|X,y]=\mathbb{P}[h]\,\mathbb{P}[y|X,h]/\mathbb{P}[y|X]\sim\exp\Big\{-\frac{1}{2\bar\sigma^2}\|h-\bar\mu\|_2^2-\frac{1}{2\sigma^2}\|y-Xh\|_2^2\Big\}.$$

Completing the square in the usual fashion shows that the posterior of learner t is given by:

$$\begin{cases}
\Sigma_t=\left(\bar\sigma^{-2}I_d+\sigma^{-2}X_t^TX_t\right)^{-1};\\
\mu_t=\Sigma_t\left(\bar\sigma^{-2}\bar\mu+\sigma^{-2}X_t^Ty_t\right),
\end{cases}\tag{23}$$

where (X_t,y_t) is the data gathered by learner t from her predecessor: specifically, $y_t=X_th+\phi_t$, where h is collected from the $(t-1)$-th learner by sampling his posterior distribution $N(\mu_{t-1},\Sigma_{t-1})$.

▶ **Theorem 5.** *Given any small enough $\delta, \varepsilon > 0$, the following sample size sequence for iterated Bayesian linear regression ensures that $\|\mathbb{E}\mu_t - \mu_0\|_2 \leq \delta$ with probability greater than $1 - \varepsilon$:*

$$m_t = D_c \frac{\|\mu_0 - \bar{\mu}\|_2}{\delta} \left(\frac{\sigma}{\bar{\sigma}}\right)^2 t^{1+c} + D_c\, d \log \frac{t+1}{\varepsilon},$$

for an arbitrarily small $c > 0$ and a constant D_c that depends only on c.

Proof. We proceed in two steps: first, we show that to keep $\mathbb{E}\mu_t$ arbitrarily close to μ_0 for all t hinges on spectral properties of certain random matrices; second, we call on known facts about the singular values of random Gaussian matrices to translate the spectral condition into a high-probability event. The proof unfolds as a series of simple relations, which we state first and then demonstrate. The first one follows directly from (23):

$$\mathbb{E}\mu_t = (I_d + M_t)^{-1} (\bar{\mu} + M_t\, \mathbb{E}\mu_{t-1}), \quad \text{where} \quad M_t := \left(\frac{\bar{\sigma}}{\sigma}\right)^2 X_t^T X_t. \tag{24}$$

Note that (24) is a randomized recursive relation since the data points $X_1, X_2, \ldots$ are themselves random. We note that all the matrices whose inverses are taken are positive definite, hence nonsingular. To move on to our second relation, we define the matrix

$$Q_t := (I_d + M_t)^{-1} M_t (I_d + M_{t-1})^{-1} M_{t-1} \cdots (I_d + M_1)^{-1} M_1,$$

for $t > 0$, with $Q_0 = I_d$, and prove by induction that

$$\mathbb{E}\,\mu_t = Q_t \mu_0 + (I_d - Q_t)\bar{\mu}. \tag{25}$$

The base case is obvious so we assume that $t > 0$: by (24),

$$\begin{aligned}
\mathbb{E}\,\mu_t &= (I_d + M_t)^{-1}(\bar{\mu} + M_t\, \mathbb{E}\,\mu_{t-1}) \\
&= (I_d + M_t)^{-1}(\bar{\mu} + M_t Q_{t-1}\mu_0 + M_t(I_d - Q_{t-1})\bar{\mu}) \\
&= (I_d + M_t)^{-1} M_t Q_{t-1}\mu_0 + (I_d + M_t)^{-1}(I_d + M_t(I_d - Q_{t-1}))\bar{\mu} \\
&= Q_t\, \mu_0 + (I_d - (I_d + M_t)^{-1} M_t Q_{t-1})\bar{\mu},
\end{aligned}$$

which proves (25). Our next goal is to bound the information decay $\|\mathbb{E}\,\mu_t - \mu_0\|_2$. To do that, we investigate the spectral norm of the matrix $I_d - Q_t$, which leads to our third relation. We prove by induction that, for $t > 0$,

$$\|I_d - Q_t\|_2 \leq \sum_{s=1}^{t} \|A_s\|_2, \tag{26}$$

where $A_s := (I_d + M_s)^{-1}$. For $t = 1$, $Q_1 = (I_d + M_1)^{-1} M_1 = I_d - (I_d + M_1)^{-1}$ and the claim follows. If $t > 1$, then

$$\begin{aligned}
\|I_d - Q_t\|_2 &= \|(I_d - Q_{t-1}) + (Q_{t-1} - Q_t)\|_2 \\
&\leq \|I_d - Q_{t-1}\|_2 + \|Q_t - Q_{t-1}\|_2 \leq \sum_{s=1}^{t-1} \|A_s\|_2 + \|\Psi\|_2,
\end{aligned}$$

where $\Psi := (A_t M_t - I_d) Q_{t-1}$. Since $A_t(I_d + M_t) = I_d$, we have $\Psi = -A_t Q_{t-1}$. Each matrix M_s is positive semidefinite, so the eigenvalues of $(I_d + M_s)^{-1} M_s$ are of the form $\lambda/(1+\lambda)$, where $\lambda \geq 0$. This shows that all the eigenvalues of Q_s are between 0 and 1;

therefore $\|Q_s\|_2 \le 1$. The eigenvalues of $I_d - A_t M_t$ are the same as those of A_t; hence, by submultiplicativity, $\|\Psi\|_2 \le \|A_t\|_2 \|Q_{t-1}\|_2 \le \|A_t\|_2$, which establishes (26).

We are now ready to express the information decay in spectral terms. Pick an arbitrarily small constant $c > 0$ and assume that

$$\|A_s\|_2 \le \frac{\delta}{\|\bar{\mu} - \mu_0\|_2} \Big(\frac{c}{1+c}\Big) \Big(\frac{1}{s}\Big)^{1+c}. \tag{27}$$

By (25), $\mathbb{E}\,\mu_t - \mu_0 = (I_d - Q_t)(\bar{\mu} - \mu_0)$; therefore, by (26),

$$\begin{aligned} \|\mathbb{E}\,\mu_t - \mu_0\|_2 &\le \|\bar{\mu} - \mu_0\|_2 \sum_{s=1}^{t} \|A_s\|_2 \le \frac{\delta c}{1+c} \sum_{s=1}^{t} s^{-1-c} \\ &\le \frac{\delta c}{1+c} \Big(1 + \int_1^{\infty} x^{-1-c}\, dx\Big) = \delta, \end{aligned} \tag{28}$$

The relation says that, on average, the means of any of the agents' posteriors can be brought as close to the original mean to be learned as we want. We can turn this into a high-probability event by using some basic random matrix theory. Recall that $\mathbb{E}\,\mu_t$ is itself a random variable whose stochasticity comes from the matrices X_s, which are all drawn from Gaussians. Because M_s is positive semidefinite,

$$\|A_s\|_2 \le \|M_s^{-1}\|_2 \le \frac{(\sigma/\bar{\sigma})^2}{\lambda_{\min}(X_t^T X_t)} \le \Big(\frac{\sigma/\bar{\sigma}}{\sigma_1(X_t)}\Big)^2, \tag{29}$$

which gives us a relation between the spectral norm of $(I_s + M_s)^{-1}$ and the smallest singular value $\sigma_1(X_t)$ of an m_t-by-d matrix X_t whose elements are drawn *iid* from $N(0,1)$. The asymptotic behavior of $\sigma_1(X_t)$ for large values of m_t has been extensively studied within the field of random matrix theory [5, 6, 17]. Following Theorem II.13 in (Davidson & Szarek [5]), for any $\gamma_t > 0$,

$$\mathbb{P}[\sigma_1(X_t) < \sqrt{m_t} - \sqrt{d} - \gamma_t] \le e^{-\gamma_t^2/2}.$$

We use C below as a generic constant large enough to satisfy the inequalities where it appears. Setting $\gamma_t = C\sqrt{\log((t+1)/\varepsilon)}$ ensures that $\sum_{t>0} e^{-\gamma_t^2/2} < \varepsilon$, hence that $\sigma_1(X_t) < \sqrt{m_t} - \sqrt{d} - \gamma_t$ holds for all t with probability less than ε. With our setting of m_t, this means that, for all $t > 0$,

$$\mathbb{P}\Big[\,\sigma_1(X_t) \ge \frac{\sqrt{m_t}}{2}\,\Big] > 1 - \varepsilon. \tag{30}$$

Assuming the event in (30), it follows from (29) and our setting of m_t that

$$\|A_t\|_2 \le \frac{4}{m_t} \Big(\frac{\sigma}{\bar{\sigma}}\Big)^2 \le \frac{\delta}{\|\bar{\mu} - \mu_0\|_2} \Big(\frac{4}{D_c}\Big) \Big(\frac{1}{t}\Big)^{1+c};$$

hence (27) for D_c large enough. By (28, 30), this proves that, with probability greater than $1 - \varepsilon$, $\|\mathbb{E}\,\mu_t - \mu_0\|_2 \le \delta$ for all $t > 0$, which completes the proof. ◀

References

1 FC Bartlett. Remembering: a study in experimental and social psychology.(1932). 317 pp.

2 Aaron Beppu and Thomas L Griffiths. Iterated learning and the cultural ratchet. In *Proceedings of the 31st annual conference of the cognitive science society*, pages 2089–2094. Citeseer, 2009.

3 A Bhattachayya. On a measure of divergence between two statistical population defined by their population distributions. *Bulletin Calcutta Mathematical Society*, 35:99–109, 1943.

4 Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 142–149. IEEE, 2000.

5 Kenneth R Davidson and Stanislaw J Szarek. Local operator theory, random matrices and banach spaces. *Handbook of the geometry of Banach spaces*, 1(317-366):131, 2001.

6 Alan Edelman. Eigenvalues and condition numbers of random matrices. *SIAM Journal on Matrix Analysis and Applications*, 9(4):543–560, 1988.

7 Thomas L Griffiths and Michael L Kalish. A bayesian view of language evolution by iterated learning. In *Proceedings of the 27th annual conference of the cognitive science society*, pages 827–832, 2005.

8 Thomas L Griffiths and Michael L Kalish. Language evolution by iterated learning with bayesian agents. *Cognitive Science*, 31(3):441–480, 2007.

9 Thomas L Griffiths, Michael L Kalish, and Stephan Lewandowsky. Theoretical and empirical evidence for the impact of inductive biases on cultural evolution. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 363(1509):3503–3514, 2008.

10 Michiel Hazewinkel. *Encyclopaedia of Mathematics*. Springer Science & Business Media, 2013.

11 Michael L Kalish, Thomas L Griffiths, and Stephan Lewandowsky. Iterated learning: Intergenerational knowledge transmission reveals inductive biases. *Psychonomic Bulletin & Review*, 14(2):288–294, 2007.

12 Simon Kirby, Tom Griffiths, and Kenny Smith. Iterated learning and the evolution of language. *Current opinion in neurobiology*, 28:108–114, 2014.

13 James R Norris. *Markov chains*. Cambridge university press, 1998.

14 Amy Perfors and Daniel Navarro. Language evolution is shaped by the structure of the world: An iterated learning analysis. In *Annual Conference*, 2011.

15 Anna N Rafferty, Thomas L Griffiths, and Dan Klein. Convergence bounds for language evolution by iterated learning. In *Proceedings of the Thirty-First Annual Conference of the Cognitive Science Society*, 2009.

16 Anna N Rafferty, Thomas L Griffiths, and Dan Klein. Analyzing the rate at which languages lose the influence of a common ancestor. *Cognitive science*, 38(7):1406–1431, 2014.

17 Mark Rudelson and Roman Vershynin. Smallest singular value of a random rectangular matrix. *Communications on Pure and Applied Mathematics*, 62(12):1707–1739, 2009.

18 Kenny Smith. Iterated learning in populations of bayesian agents. In *Proceedings of the 31st annual conference of the cognitive science society*, pages 697–702. Citeseer, 2009.

19 Mónica Tamariz and Simon Kirby. Culture: copying, compression, and conventionality. *Cognitive science*, 39(1):171–183, 2015.

A Appendix

The two forms of the function d_{RS} in (3) make it clear that $0 \leq d_{RS}(\boldsymbol{a}, \boldsymbol{b}) \leq 1$ and $d_{RS}(\boldsymbol{a}, \boldsymbol{b}) = 0$ if and only if $\boldsymbol{a}$ and $\boldsymbol{b}$ are identical. We easily check that d_{RS} makes the simplex $\mathcal{S}$ of distributions over $\mathcal{D}$ into a metric space. Indeed, $d_{RS}(\cdot, \cdot)$ is obviously symmetric, and $d_{RS}(\boldsymbol{a}, \boldsymbol{b}) = 0$ implies that $\boldsymbol{a} = \boldsymbol{b}$. To check the triangular inequality, notice that

$$d_{RS}(\boldsymbol{a}, \boldsymbol{b}) = \sqrt{1 - \Big(\sum_{i=1}^{s} \sqrt{a_i b_i}\Big)^2} = \sin\langle \sqrt{\boldsymbol{a}}, \sqrt{\boldsymbol{b}} \rangle, \tag{31}$$

where $\langle\sqrt{\boldsymbol{a}},\sqrt{\boldsymbol{b}}\,\rangle$ is the angle between the unit vectors $\sqrt{\boldsymbol{a}}$ and $\sqrt{\boldsymbol{b}}$, using the notation $\sqrt{\boldsymbol{v}} = (\sqrt{v_1},\ldots,\sqrt{v_s})$. To prove that $d_{RS}(\boldsymbol{a},\boldsymbol{b}) + d_{RS}(\boldsymbol{b},\boldsymbol{c}) \geq d_{RS}(\boldsymbol{a},\boldsymbol{c})$ for any $\boldsymbol{a},\boldsymbol{b},\boldsymbol{c} \in \mathcal{S}$, we denote by α,β,γ the corresponding angles in that order, ie, $\alpha = \langle\sqrt{\boldsymbol{a}},\sqrt{\boldsymbol{b}}\,\rangle$, etc. The coordinates in $\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}$ are nonnegative; therefore $0 \leq \alpha,\beta,\gamma \leq \pi/2$. These form the three angles at the origin of a tetrahedron with a vertex at the origin; therefore, by the triangular inequality in spherical geometry, $\alpha+\beta \geq \gamma$. If $\alpha+\beta \leq \frac{\pi}{2}$, then $\sin\alpha+\sin\beta \geq \sin\alpha\cos\beta+\cos\alpha\sin\beta = \sin(\alpha+\beta) \geq \sin\gamma$. On the other hand, if $\alpha+\beta > \pi/2$, then $\sin\alpha+\sin\beta = 2\sin\frac{\alpha+\beta}{2}\cos\frac{\alpha-\beta}{2} \geq 2\sin\frac{\pi}{4}\cos\frac{\pi}{4} = 1 \geq \sin\gamma$, which establishes the triangular inequality.

Relation to the Euclidean distance

Shrinking the simplex $\mathcal{S}$ by a tiny amount, we define $\mathcal{S}_\varepsilon := \{\boldsymbol{a} \in \mathcal{S} : \varepsilon \leq a_i \leq 1-\varepsilon\}$ and note that

$$d_E(\boldsymbol{a},\boldsymbol{b}) := \|\boldsymbol{a}-\boldsymbol{b}\|_2 = \sqrt{\sum_{i=1}^{s}(\sqrt{a_i}-\sqrt{b_i})^2(\sqrt{a_i}+\sqrt{b_i})^2}.$$

It follows that, for $\boldsymbol{a},\boldsymbol{b} \in \mathcal{S}_\varepsilon$,

$$\frac{1}{2}d_E(\boldsymbol{a},\boldsymbol{b}) \leq d_E(\sqrt{\boldsymbol{a}},\sqrt{\boldsymbol{b}}\,) \leq \frac{1}{2\sqrt{\varepsilon}}\,d_E(\boldsymbol{a},\boldsymbol{b}). \tag{32}$$

On the other hand, $\|\sqrt{\boldsymbol{a}}\|_2 = \|\sqrt{\boldsymbol{b}}\|_2 = 1$, so the vectors $\sqrt{\boldsymbol{a}}$ and $\sqrt{\boldsymbol{b}}$ form an isosceles triangle; hence

$$d_E(\sqrt{\boldsymbol{a}},\sqrt{\boldsymbol{b}}\,) = 2\sin\frac{1}{2}\langle\sqrt{\boldsymbol{a}},\sqrt{\boldsymbol{b}}\rangle = \frac{\sin\langle\sqrt{\boldsymbol{a}},\sqrt{\boldsymbol{b}}\rangle}{\cos\frac{1}{2}\langle\sqrt{\boldsymbol{a}},\sqrt{\boldsymbol{b}}\rangle} = \frac{d_{RS}(\boldsymbol{a},\boldsymbol{b})}{\cos\frac{1}{2}\langle\sqrt{\boldsymbol{a}},\sqrt{\boldsymbol{b}}\rangle}\,.$$

Since $0 \leq \langle\sqrt{\boldsymbol{a}},\sqrt{\boldsymbol{b}}\,\rangle \leq \frac{\pi}{2}$,

$$d_{RS}(\boldsymbol{a},\boldsymbol{b}) \leq d_E(\sqrt{\boldsymbol{a}},\sqrt{\boldsymbol{b}}\,) \leq \sqrt{2}\,d_{RS}(\boldsymbol{a},\boldsymbol{b}).$$

Together with (32) this shows that, for any $\boldsymbol{a},\boldsymbol{b} \in \mathcal{S}_\varepsilon$,

$$\frac{1}{2\sqrt{2}}\,d_E(\boldsymbol{a},\boldsymbol{b}) \leq d_{RS}(\boldsymbol{a},\boldsymbol{b}) \leq \frac{1}{2\sqrt{\varepsilon}}\,d_E(\boldsymbol{a},\boldsymbol{b}), \tag{33}$$

which shows that the Euclidean distance and the metric d_{RS} are equivalent in $\mathcal{S}_\varepsilon$.

Relation to other distances

The metric d_{RS} is related to the Hellinger and Bhattacharyya distances. Writing $C(\boldsymbol{a},\boldsymbol{b}) = \sum_{i=1}^{s}\sqrt{a_ib_i}$ [4], then $d_{RS}(\boldsymbol{a},\boldsymbol{b}) = \sqrt{1-C(\boldsymbol{a},\boldsymbol{b})^2}$. The Hellinger distance is defined as $d_H(\boldsymbol{a},\boldsymbol{b}) = \sqrt{1-C(\boldsymbol{a},\boldsymbol{b})}$ [10], while the Bhattacharyya distance is defined as $d_B(\boldsymbol{a},\boldsymbol{b}) = -\ln C(\boldsymbol{a},\boldsymbol{b})$ [3]. The total variation distance d_{TV} is half the ℓ_1-norm; therefore $d_{TV}(\boldsymbol{a},\boldsymbol{b}) \leq \frac{1}{2}\sqrt{s}\,d_E(\boldsymbol{a},\boldsymbol{b})$. Combining these observations with (33) establishes (4):

$$\begin{cases} d_H = \sqrt{1-\sqrt{1-d_{RS}^2}}\,; \\ d_B = -\frac{1}{2}\ln(1-d_{RS}^2)\,; \\ d_{TV} \leq \sqrt{2s}\,d_{RS}. \end{cases}$$

Coding in Undirected Graphs Is Either Very Helpful or Not Helpful at All

Mark Braverman[*1], **Sumegha Garg**[2], **and Ariel Schvartzman**[3]

1 **Princeton University, Princeton, USA**
mbraverm@princeton.edu
2 **Princeton University, Princeton, USA**
sumeghag@cs.princeton.edu
3 **Princeton University, Princeton, USA**
acohenca@cs.princeton.edu

Abstract

While it is known that using network coding can significantly improve the throughput of directed networks, it is a notorious open problem whether coding yields *any* advantage over the multicommodity flow (MCF) rate in undirected networks. It was conjectured in [11] that the answer is 'no'. In this paper we show that even a small advantage over MCF can be amplified to yield a near-maximum possible gap.

We prove that any undirected network with k source-sink pairs that exhibits a $(1+\varepsilon)$ gap between its MCF rate and its network coding rate can be used to construct a family of graphs G' whose gap is $\log(|G'|)^c$ for some constant $c < 1$. The resulting gap is close to the best currently known upper bound, $\log(|G'|)$, which follows from the connection between MCF and sparsest cuts.

Our construction relies on a gap-amplifying graph tensor product that, given two graphs G_1, G_2 with small gaps, creates another graph G with a gap that is equal to the product of the previous two, at the cost of increasing the size of the graph. We iterate this process to obtain a gap of $\log(|G'|)^c$ from any initial gap.

1998 ACM Subject Classification G.2.2. Graph Theory, Network problems

Keywords and phrases Network coding, Gap Amplification, Multicommodity flows

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.18

1 Introduction

The area of network coding addresses the following basic problem: in a distributed communication scenario, can one use coding to outperform packet routing-based solutions? While the problem of communicating information over a network can be viewed as the process of moving information packets between terminals, a key distinction between moving packets and moving commodities is that information packets can be re-encoded by intermediate nodes. For example, a node which receives packets P_1 and P_2 can calculate and transmit the bitwise XOR packet $P_1 \oplus P_2$ to its neighbor. This operation has no analogue in multicommodity flow scenarios.

* Research supported in part by an NSF Awards, DMS-1128155, CCF-1525342, and CCF-1149888, a Packard Fellowship in Science and Engineering, and the Simons Collaboration on Algorithms and Geometry. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

© Mark Braverman, Sumegha Garg, and Ariel Schvartzman;
licensed under Creative Commons License CC-BY

Whether (and to what extent) this ability confers any benefits over the simple routing-based solution, depends on the specific goal of the communication at hand. Such goals may include uni-cast and multi-cast throughput, error-resilience and security, to name a few. These questions have been the subject of active study in the recent past. A summary of major directions can be found in the books [14, 13], and surveys [5, 15].

In this paper we focus on noiseless unicast communication. The network is a capacitated graph G with k source-sink terminal pairs (s_i, t_i). Each source vertex s_i wants to transmit an information stream to t_i. The network coding rate $\mathrm{NC}(G)$ is the maximum rate at which transmission between all pairs can happen simultaneously, given the capacity constraints [2].

If we forbid coding, and restrict nodes to forwarding information packets that they receive, the problem becomes equivalent to multicommodity flow over G — the very well-studied problem of maximizing the rate $\mathrm{MCF}(G)$ at which commodities are moved from sources to sinks subject to the capacity constraints (see e.g. [3] for background). Clearly, the multicommodity rate can always be achieved — but can it be beaten using "bit tricks"?

If the graph G is directed, there are well-known examples which show that coding can improve throughput in a very dramatic way [7, 1]. There is a family of examples G, where the gap between the multicommodity flow rate and network coding throughput is as large as $O(|G|)$. Despite substantial effort, it is not clear whether coding confers *any* benefit over routing in undirected networks. Li and Li [11] conjectured that the answer is 'no'. This conjecture is currently open.

It is known that the Li and Li conjecture holds in some special cases. Naturally, it holds whenever the sparsity of the graph matches the multicommodity flow rate. For cases where these quantities are not equal, [8] and [9] show that the conjecture is true for the Okamura-Seymour graph and [8, 1] show it for an infinite family of bipartite graphs. Empirical evidence also suggests that the conjecture is true [12].

One simple case where coding rate cannot exceed capacity is the case when the channel is a single edge: two parties cannot be sending messages to each other at a total rate exceeding the channel's capacity. This is a simple consequence of Shannon's Noiseless Coding Theorem. As a simple corollary, the sparsest cut in G provides an upper bound for the network coding rate $\mathrm{NC}(G)$. The sparsity of a cut $(U, V \setminus U)$ is defined as

$$\mathrm{Sparsity}(U, V \setminus U) := \frac{\mathrm{Capacity}(U, V \setminus U)}{\mathrm{Demand}(U, V \setminus U)} \tag{1}$$

If we merge the vertices on either side of the cut, the network coding rate becomes $\mathrm{Sparsity}(U, V \setminus U)$. Merging nodes can only increase network coding rate, and thus we have $\mathrm{NC}(G) \leq \mathrm{Sparsity}(U, V \setminus U)$. Since the sparsity of G is defined as the minimum of (1) over all cuts, we have $\mathrm{NC}(G) \leq \mathrm{Sparsity}(G)$.

As discussed below, the multicommodity flow problem is very well-studied. In the one commodity case, the Max-Flow Min-Cut Theorem asserts that sparsity is equal to the flow rate. In the multicommodity case, the sparsity is still an upper bound on the multicommodity flow rate $\mathrm{MCF}(G)$, but it might be loose by a factor of $\log |G|$:

$$\mathrm{Sparsity}(G)/O(\log |G|) \leq \mathrm{MCF}(G) \leq \mathrm{Sparsity}(G). \tag{2}$$

Thus the advantage one can gain for network coding over undirected graphs is at most $O(\log |G|)$:

$$\mathrm{NC}(G)/O(\log |G|) \leq \mathrm{Sparsity}(G)/O(\log |G|) \leq \mathrm{MCF}(G) \leq \mathrm{NC}(G). \tag{3}$$

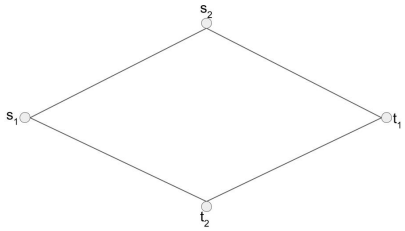


Figure 1 Graph G_1 and G_2.

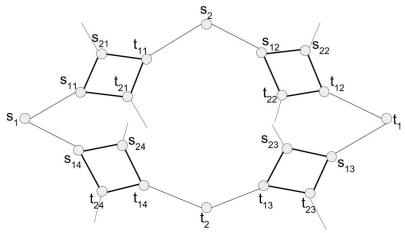


Figure 2 Basic gadget that embeds a copy of G_2 into each edge of G_1. The edges coming out of the copies will be used to connect to other copies of the outer graph (G_1). Labelled source-sink pairs of G_2 are just for reference. The thin edges are just an artifact and their respective end points represent a single vertex.

The Li and Li conjecture asserts that the rightmost '$\leq$' is indeed an equality. Our main result is that either the conjecture is true, or it must be nearly 'completely false': the gap between $\mathrm{NC}(G)$ and $\mathrm{MCF}(G)$ can be as high as poly-logarithmic in $|G|$.

▶ **Theorem 1.** *Given a graph G that achieves a gap of $1+\epsilon$ between the multicommodity flow rate and the network coding rate, we can construct an infinite family of graphs $\widetilde{G}$ that achieve a gap of $O\left(\log|\widetilde{G}|\right)^c$ for some constant $c<1$ that depends on the original graph G.*

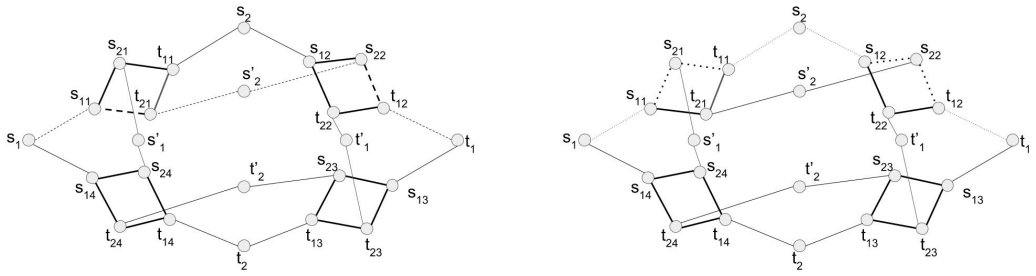
In order to prove Theorem 1, we will show a simpler construction that can be applied repeatedly.

▶ **Theorem 2.** *Given a graph G of size n with a gap of $1+\epsilon$ between the multicommodity flow rate and the network coding rate, we can create another graph G' of size n^{c^2} and a gap of $(1+\epsilon)^2$, where c depends on the diameter of the graph G.*

The idea of gap-amplification in the context of network coding has been studied before. In [4] the authors combine linear programming techniques with hypergraph product operations to prove general gap amplification results between linear and non-linear network coding for directed graphs.

Proof outline

The main part of the construction is to define a *graph tensor* on graphs G_1 and G_2 that have gaps of $1+\epsilon_1$ and $1+\epsilon_2$ respectively, between the multicommodity flow rate and the network coding rate, which gives a new graph G with a gap of $(1+\epsilon_1)(1+\epsilon_2)$ while keeping a check on the size of G. We can then take a graph with a small gap and tensor it with itself to produce a graph with an even larger gap. Repeatedly tensoring the output of the previous iteration with itself will give us Theorem 1.



(a) A cheating path is highlighted with dashed edges.

(b) An honest path is highlighted with dotted edges.

Figure 3 Two copies of G_1 (with source-sink pairs $s_1 - t_1, s_2 - t_2$ and $s'_1 - t'_1, s'_2 - t'_2$, respectively) have 8 edges. These edges are replaced with 4 copies of G_2, each copy having two source-sink pairs. All edges in the first copy of G_1 are replaced with s_{1X}-t_{1X} source-sink pairs of G_2; edges in the second copy of G_2 are replaced with s_{2X}-t_{2X} source-sink pairs of G_2

In G_2, network coding allows us to send more information from every source to its corresponding sink than what simple flows allow. We construct a gadget for the graph tensor exploiting this fact as in Figure 2. We replace each edge of G_1 by a copy of G_2 with endpoints at a deterministic source-sink pair. We keep the source-sink pairs of G_1 and edges of G_2.

For simplicity, assume that each edge in Figure 1 has capacity 1. The effective capacity at each edge seen by G_1 under network coding is more than that under flows. Intuitively, replacing each edge with a source-sink pair of G_2 should give network coding a "capacity advantage" of $(1 + \epsilon_2)$ over multicommodity flow. Since the information transferred grows linearly with the capacity, the new information exchanged between source-sink pairs in the gadget under network coding should be $(1 + \epsilon_1)(1 + \epsilon_2)$ times the information exchanged under flows.

We need to be careful because G_2 exhibits a gap only when we need to send information from all sources simultaneously. We address this by adding more copies of G_1 to the graph tensor and replacing its edges with other source-sink pairs of the copies of G_2 as in Figure 3. In each copy of G_1, we replace all edges with the same source-sink pair of G_2. At the same time, each copy of G_2 serves to replace the same edge in all copies of G_1. This is done to facilitate the proof of the upper bound on the MCF rate in the resulting graph.

Our work is not done here. It is easy to get a lower bound on the network coding rate on the final graph G by just showing a network coding solution. For this, informally, we just compose the network coding solutions for G_1 and G_2. The hard part is to get an upper bound on the multicommodity flow rate. Since MCF is a linear program, we can upper bound the value of multicommodity flows by looking at the dual solution of its relaxed linear program. This dual, described in Section 2, involves computing shortest distances between source-sink pairs under some metric. This metric readily tensorizes: we can take the length of an edge in a copy of G_2 to be the product of the length of that edge in G_2 times the length of the edge(s) in G_1 this particular copy of G_2 is replacing. The problem is to get the lengths of *whole paths* to tensorize.

What could go wrong? Consider Figure 3. We would ideally want the dotted paths as in Figure 3b to be the shortest path between $S1$ and $T1$ in G', since its length is the length of the shortest $s_1 - t_1$ path in G_1 times the length of the shortest $s_{1X} - t_{1X}$ path in G_2. Unfortunately, during the tensoring operation we inadvertently introduce additional $s_1 - t_1$ paths that do not correspond to "products" of paths from G_1 and G_2. For example, the dashed path in Figure 3a is a "cheating" path which can make the distance between s_1 and t_1 shorter than expected. We deter the use of "cheating" paths by increasing the number of

hops between different copies of G_1 that a path has to take before it reaches the same copy again. The technical ingredient which prevents such cheating is in the design of the bipartite graph which tells which copy of G_1 should use which copies of G_2 (and how to connect them). To prevent cheating, the bipartite graph will need to be of high girth. The crucial part of the construction is thus constructing high girth bipartite graphs while still keeping check on the size so as to get a $O(\log(\text{size})^c)$ $(c < 1)$ gap when the tensor is applied repeatedly.

Discussion

A natural question arises: Can we have a tensor construction that starts with a graph G having some gap between the multicommodity flow rate and the network coding rate and outputs a graph G' with gap $\omega(\log(|G'|))$, thus contradicting (3) and proving the Li and Li conjecture? We address this question with respect to our construction in Section 4. We show that the MCF vs. Sparsest Cut gap tensorizes for our construction, and thus the tensorization process on its own cannot cause the gap to exceed $O(\log |G'|)$.

At the same time, if one's goal is to prove the conjecture, it might be easier to reach a contradiction to the gap being $(\log |G|)^c$ than to a constant gap.

2 Preliminaries

In this section we introduce the problems that we will be interested in studying and any relevant notation. Where appropriate, we use the same notation and definitions as [1, 7].

When $G = (V, E)$ is a graph, we specify vertex set of G with $V(G)$ when the underlying graph G is not clear from the context. Similarly $E(G)$ represents the edge set for graph G. The set $\{1, 2, ..., n\}$ is represented by $[n]$. $I(G)$ denotes the set of k source-sink pairs $(s_i, t_i), i \in [k], s_i, t_i \in V$. Given a bipartite graph $B = (V_1, V_2, E)$, we denote the left side of the graph by $V_1(B)$ and the right by $V_2(B)$. A bipartite graph is (r, s) bi-regular when each vertex on the left side has degree r whereas each vertex on right side has degree s.

2.1 Network coding

▶ **Definition 3.** An instance of the k-pairs communication problem consists of

- a graph $G = (V, E)$,
- a capacity function $c : E \to \mathbb{R}^+$,
- a set I of commodities of size k, each of which can be described by a triplet of values (s_i, t_i, d_i) corresponding to the source node, the sink node and the demand of commodity i.

In line with [1], for undirected graphs we consider each edge e as two directed edges $\vec{e}, \overleftarrow{e}$, whose capacities will be defined later. It will also be convenient to think of source and sink nodes as edges. Therefore, for every source and sink pair (s_i, t_i), we create new nodes S_i, T_i and connect them via single edges to s_i, t_i respectively. These edges are of unbounded capacity and we will refer to these as the source and sink edges respectively. Every source S_i wants to communicate a message to its sink.

We give the formal definition of a network coding solution in Appendix A. Let M_i be the set of messages the i-th source-sink pair wants to communicate, and $M = \prod_i M_i$. Let $\Delta(e)$ be the alphabet of characters available at edge e. Informally, the solution to a network coding problem must specify for each edge e a function $f_e : M \to \Delta(e)$, which dictates the character transmitted on that edge. The function f_e must be computable from the characters on the incoming edges at the sender end point. The message at the source and sink edges of any commodity must agree.

The network coding rate (henceforth known as coding rate) is the largest value r such that for each source-sink pair at least $r \cdot d_i$ information is transmitted while preserving the capacity constraint on all edges.

2.2 Multicommodity flow problems and sparsity cuts

A flow problem consists of a graph $G = (V, E)$ with k commodities together with k pairs of nodes (s_i, t_i) and quantities d_i. The goal is to transmit d_i units of commodity i from s_i to t_i while keeping the total sum of commodities that go through a given edge e below its capacity $c(e)$. There are many optimization problems surrounding this problem. We will focus on the following one: what is the largest λ such that at least λ fraction of each commodity's demand is routed? This is justified by assuming that no commodity is prioritized over another and that all resources are shared. We refer to this quantity as the flow rate of the graph. There are well-known linear programming formulations for these problems (see LP 5 in Section B in the appendix). Since we will be interested in providing provable upper bounds to the flow rate, it will suffice to look at the dual of this problem. In particular, we use the variables on the following dual LP to provide upper bounds on the flow rate of the sequence of graphs we create. We will refer to the $w_{(u,v)}$ as the weight of edge (u, v) in the dual solution.

$$\begin{aligned} \text{minimize} \sum_{u,v} w_{(u,v)} c(u,v) & \\ \text{subject to} \sum_{(s_i,t_i)} l(s_i,t_i) d_i \geq 1 & \qquad \text{(Distance Constraint)} \\ \sum_{(u,v)\in p} w_{(u,v)} \geq l(s_i,t_i) & \qquad \forall i \in [k], p \in P_i \\ w_{(u,v)} \geq 0 & \qquad \forall (u,v) \in E \wedge \forall (u,v) = (s_i,t_i) \end{aligned} \tag{4}$$

This LP introduces a semi-metric on the graph which assigns weights to the edges. $l(s_i, t_i)$ is the shortest distance between $i-$th source-sink pair w.r.t. this metric. The goal is to minimize the weighted length of the edges of the graph while maintaining a certain separation between the source-sink pairs. Zero weight edges can be problematic for our graph tensor since they may reduce the weighted girth of the graph in ways we cannot account for. Our tensor, however, does not produce new zero weight edges. Therefore it suffices for our purposes to show that we can get rid of them at the beginning of the construction.

▶ **Lemma 4.** *If G is a graph such that the gap between the flow rate and the coding rate is $(1 + \epsilon)$, a new graph G' can be constructed such that the gap does not decrease and all the edge weights in LP (4) are non-zero.*

Proof. We defer the proof of this lemma to Section B of the appendix. ◀

Interestingly, this lemma is not true for directed graphs.

3 Construction

In this section we present the construction of our graph tensor and prove our main results, Theorems 1 and 2. The construction takes two graphs with small gaps and tensors them in such a way that the resulting graph has a gap equal to the product of the previous gaps. Iteratively tensoring a graph with a small gap with itself will yield our main results.

Throughout this section, when referring to a graph $G_i = (V_i, E_i), i \in [2]$, k_i is the number of source-sink pair, v_i the number of vertices and m_i the number of edges. The capacity of edge $e \in E_i$ will be denoted by c_{ie}.

3.1 Overview

As mentioned in Section 1, we need a bijection between the graph tensor on G_1 and G_2 and bipartite graphs. We represent the copies of G_1 by numbered nodes on the left side of the bipartite graph (say B) and copies of G_2 by nodes on the right side of B. We add an edge (i, j) in B when an edge in the i-th copy of G_1 got replaced by the j-th copy of G_2 aligned at a specific source-sink pair. But this definition of bipartite graph B loses information about which specific edge was replaced with which specific source-sink pair. Thus, we consider a variant of bipartite graphs: *colored bipartite graphs*, which have two colors associated with each edge. We will use the first color to represent the edge that got replaced in a copy of G_1 and the other to represent the source-sink pair of G_2 that replaced that edge. Thus, edges of B get colored from the set $[m_1] \times [k_2]$. Note that each vertex on the left side has degree m_1 and that on right hand side has degree k_2. The formal description of colored bipartite graphs and *graph tensor* based on this idea is given in Subsection 3.2.

As discussed in Section 1, we can avoid "cheating" paths by increasing the number of hops that a dashed path (Figure 3) needs to take to come back to the same copy of G_1. Our first requirement would be for the colored bipartite graph B to have high girth. Lemma 19 states the existence of nearly optimal sized high girth bipartite graphs and Subsection 3.2.2 shows how to construct specific colored bipartite graphs (as in Subsection 3.2) of high girth.

Is having a high girth B sufficient for the number of hops to be large? No. When G_2 has two sources at the same vertex, the end points (on source side) of the edges in copies of G_1 that these two source-sink pairs replaced will collapse on the same vertex implying that we can move between these copies of G_1 instantly without traveling along any edge in the tensored graph. But, we would have travelled two consecutive edges in B. To remedy this, we condition on the graph G_2 to have all sources and sinks lying on distinct vertices. Note that the length of the cheating paths is defined with respect to the weights of edges in a dual solution. Thus, we cannot just transfer the source/sinks to leaves at the corresponding vertex through infinite capacity edges as they would always get weight 0 in the dual. In Subsection 3.2.1, we present a way to modify graph G_2 to satisfy the above condition.

The multicommodity flow rate for the tensored graph is upper bounded by constructing a dual solution for it based on dual solutions for graphs G_1 and G_2. In Subsection 3.2.3, we show the dual construction and prove that the gap of the tensored graph is the product of the previous gaps given appropriate girth.

The last subsection of this section contains the details of repeated tensoring to get Theorem 1.

3.2 Graph Tensor

▶ **Definition 5. Colored Bipartite Graph:** We define $\mathcal{B}_{n_1,n_2,d_1,d_2,g,q_1,q_2}$ to be the set of bipartite graphs (V_1, V_2, E) with girth g, $|V_1| = n_1, |V_2| = n_2$, such that degree of each vertex in V_1 and V_2 is d_1 and d_2 respectively and each edge is given a color l_e in $[q_1] \times [q_2]$. Note that $n_1 d_1 = n_2 d_2$.

▶ **Definition 6.** $\boldsymbol{T(G_1, G_2, B)}$ is defined to be the graph tensor on directed graphs G_1 and G_2 based on the colored bipartite graph B.

For $T(G_1, G_2, B)$ to be defined, we need B to satisfy the following properties:

1. $B \in \mathcal{B}_{n_1,n_2,m_1,k_2,g,m_1,k_2}$ for some $n_1, g \in \{1, 2, ...\}$.
   - G_1 has m_1 edges and G_2 has k_2 source-sink pairs. Therefore the degrees of each node on the left and right sides should be m_1 and k_2, respectively.
   - As mentioned in Subsection 3.1, edges must be colored in the set $[m_1] \times [k_2]$.
2. $\forall v \in V_2$, the set $B_v = \{b_e \mid e \text{ is incident to } v \text{ and } l_e = (a_e, b_e)\}$ is the complete set $[k_2]$. We want each source-sink pair of a copy of G_2 to replace some edge in a copy of G_1.
3. $\forall u \in V_1$, the set $A_u = \{a_e \mid e \text{ is incident to } u \text{ and } l_e = (a_e, b_e)\}$ is the complete set $[m_1]$. This ensures that each edge in a copy of G_1 is replaced.
4. $\forall v \in V_2$, the set $A_v = \{a_e \mid e \text{ is incident to } v \text{ and } l_e = (a_e, b_e)\}$ has cardinality 1. To define capacities in the new tensored graph naturally, we want that each source-sink pair in a copy of G_2 replaces some unique edge in its corresponding copy of G_1.
5. $\forall u \in V_1$, the set $B_u = \{b_e \mid e \text{ is incident to } u \text{ and } l_e = (a_e, b_e)\}$ has cardinality 1. This ensures that each edges in a copy of G_1 is replaced by the same source-sink pair in different copies of G_2.

We construct the graph $T(G_1, G_2, B)$ as follows:

- Enumerate the n_1 nodes in $V_1(B)$ and n_2 nodes in $V_2(B)$: $u^{(1)}, u^{(2)}, ..., u^{(n_1)}$ and $v^{(1)}, .., v^{(n_2)}$ respectively.
- Enumerate all the edges in G_1: $e^{(1)}_{G_1}, e^{(2)}_{G_1}, ..., e^{(m_1)}_{G_1}$.
- Create n_1 copies of G_1 (vertices and source-sink pairs) and n_2 copies of G_2 (vertices and edges). Represent the x^{th} copy of graph $G_y, y \in \{1, 2\}$ by $G^{(x)}_y$. Let $u^{(i)} \in V_1(B)$ represent the i-th copy of G_1 and $v^{(j)} \in V_2(B)$ represent the j-th copy of G_2.
- For every edge $e = (u^{(i)}, v^{(j)})$ colored (p, k), merge the vertices $a_{G^{(i)}_1}$ and $s_{kG^{(j)}_2}$, and $t_{kG^{(j)}_2}$ and $b_{G^{(i)}_1}$ in $T(G_1, G_2, B)$. Here, $e^{(p)}_{G^{(i)}_1} = (a_{G^{(i)}_1}, b_{G^{(i)}_1})$ is the p^{th} edge in the i-th copy of G_1 and $(s_{kG^{(j)}_2}, t_{kG^{(j)}_2})$ is the k^{th} source-sink pair of the j^{th} copy of G_2. Informally, we are replacing each edge in a copy of G_1 by a copy of G_2 with end points aligned with the k^{th} source-sink pair. Set the capacity of every edge e' in this j^{th} copy of G_2 to be $c_{1e^{(p)}_{G_1}} c_{2e'}$. This can be done consistently due to Property (4).
- Make all the edges undirected.

We define a tensor on directed graphs to allow for composition of network coding solutions of G_1 and G_2. The direction of an edge in G_1 tells us how to align the source-sink pair of G_2 on that edge. An example of a tensor is the graph in Figure 3.

3.2.1 Standard Forms and Graph Extensions

Without loss of generality, we assume that for the graph G, all the demands $d_i, i \in [|I(G)|]$ are equal. Otherwise, we can just divide the demands into small demands of size x such that x divides all the initial rational demands. As discussed in Subsection 3.1, we want all sources and sinks to lie on distinct vertices. For all the dual solutions D that we mention, we assume that D does not contain any zero weight edges. This is justified by Lemma 4 and the fact that new duals constructed while tensoring, which will be defined later, don't create zero weight edges. We say a graph-dual pair (G, D) is in *standard form* when all the assumptions above are satisfied.

We now present a construction whose goal is to make all $s_i, t_i, i \in [k]$ lie on distinct vertices.

▶ **Definition 7.** Given a graph $G = (V, E)$ with all demands being equal to d, and a dual solution D with $\frac{NC_G}{z(D)} \geq 1 + \varepsilon$, $\forall \alpha, 0 < \alpha < \varepsilon$, construct a new graph G_α such that all

$s_i, t_i, i \in [k]$ lie on distinct vertices and G_α has a dual solution $D_\alpha(G)$ with $\frac{NC_{G_\alpha}}{z(D_\alpha(G))}$ being at least $\frac{1+\varepsilon}{1+\alpha}$. G_α is defined as the **α-Extension of G given D**. $z(D)$ is the objective value of dual solution D.

Here, we just move the sources/sinks at a vertex to the leaves of the new edges added at this vertex while keeping edge capacities and dual weights in check. The detailed description of G_α and $D_\alpha(G)$ is given in Section C of the Appendix.

3.2.2 Colored Bipartite Graph Construction

We need small, colored bipartite graphs for every degree and girth to define the graph tensor on any two graphs with gaps. We construct such graphs using biregular bipartite graphs with high girth. The following lemma states the existence of nearly-optimal sized colored bipartite graphs.

▶ **Lemma 8.** *$\forall r, s, g \geq 3$, there exists a colored bipartite graph $C_{rsg} \in \mathcal{B}_{n_1,n_2,r,s,2g,r,s}$ with $n_1, n_2 \leq (9rs)^{g+3}$.*

Proof. We defer the detailed construction and proof of the next lemma to Section D of the Appendix. ◀

3.2.3 Gap Amplification

We are given G_1 and G_2 in standard form with G_y, $y \in [2]$ having gap $(1 + \varepsilon_y)$. Let N_y be the optimal network coding solution for $G_y, y \in [2]$. Construct a directed graph G'_1 from G_1 by replacing each (undirected) edge $e = (u, v) \in E(G_1)$ of capacity c_{1e} with 2 directed edges (u, v) and (v, u) of capacities c_{1eu} and c_{1ev} respectively. Here, c_{1eu} and c_{1ev} are the capacities of edge e used by N_1 in the defined directions. Note that $c_{1eu} + c_{1ev} \leq c_{1e}$. Without loss of generality, assume $c_{1eu} + c_{1ev} = c_{1e}$, as we can always increase one of the capacities without changing the network coding solution to get the equality. Similarly, construct G'_2 from G_2 based on N_2. G'_1 and G'_2 has $m'_1 = 2m_1$ and $m'_2 = 2m_2$ edges respectively.

▶ **Definition 9. Tensor(G_1, G_2, D_1, D_2)** is defined as $T(G'_1, G'_2, B')$, where $B' = C_{m'_1 k_2 g}$, $2g = \frac{2l_1 l_2}{w_1 w_2}$. Here, l_1 and l_2 are the maximum dual distances between any source-sink pair in the dual solutions D_1 of G_1 and D_2 of G_2 respectively. $w_1 > 0$ and $w_2 > 0$ are the minimum edge weights in the dual D_1 and D_2 respectively.

Define **Dual(G_1, G_2, D_1, D_2)** to be the specific dual solution for Tensor(G_1, G_2, D_1, D_2) that would be constructed in proof of Lemma 12.

All the demands in the graph Tensor(G_1, G_2, D_1, D_2) are equal to $\frac{d_1 d_2}{q}$ where $q = \frac{V_1(B')}{k_2} = \frac{V_2(B')}{m_1}$. Here, d_y is the demand of each commodity in graph $G_y, y \in [2]$. We use such a scaling to have a simple description of Dual(G_1, G_2, D_1, D_2) in terms of D_1 and D_2.

We prove the gap amplification part of Theorem 2 next. The details of how size grows are in the next subsection.

▶ **Theorem 10.** *Given graphs G_1 and G_2 in standard form and dual solutions D_1 and D_2 respectively, such that $\frac{NC_{G_y}}{z(D_y)} \geq 1 + \varepsilon_y, y \in [2]$, $G =$ Tensor(G_1, G_2, D_1, D_2) has a dual solution $D =$ Dual(G_1, G_2, D_1, D_2) such that $\frac{NC_G}{z(D)} \geq (1 + \varepsilon_1)(1 + \varepsilon_2)$.*

In the next two lemmas, we lower bound the network coding rate and upper bound the multicommodity flow rate of G.

▶ **Lemma 11.** *The coding rate for G is at least $r_1 r_2 (1+\varepsilon_1)(1+\varepsilon_2)q$ where r_1 and r_2 are objective values of dual solutions D_1 and D_2 respectively.*

Proof Sketch. The proof follows from composing the optimal network coding solutions of G_1 and G_2. The details are given in Section E of the Appendix. ◀

▶ **Lemma 12.** *D has objective value at most $r_1 r_2 q$ where r_1 and r_2 are the objective values of dual solutions D_1 and D_2 respectively.*

Proof Sketch. $G = T(G_1', G_2', B')$ where variables are as defined in Definition 9. For every edge $e \in E(G)$, e is the undirected version of an edge in a copy of G_2' (of say e_2 in G_2') and this copy of G_2' must have replaced a unique edge (say $e_1 \in E(G_1')$) in different copies of G_1'. Edges e_1 and e_2 are directed edges but have undirected counterparts in G_1 and G_2. Let w_{1e_1} and w_{2e_2} be the weights given to the counterpart edges of e_1 and e_2 in dual solutions D_1 and D_2 respectively. Give weight $w_e = w_{1e_1} w_{2e_2}$ to edge e in D. Note that $\forall e, w_e > 0$ if $w_{1e_1}, w_{2e_2} > 0 \forall e_1, e_2$. Thus, non-zero dual solutions D_1 and D_2 give a non-zero dual solution D to graph G. We still need to show that D is a valid dual solution for G. Since B' has girth at least $\frac{2l_2 l_1}{w_1 w_2}$ and G_2 is in standard form, the dotted paths (as in Figure 3) are the shortest paths with respect to dual D. We can then write the distances between source-sink pairs in G in terms of the distance of this source-sink pair in G_1 w.r.t. D_1 and the distance of the source-sink pair in G_2 that replaced edges in this copy of G_1 w.r.t. D_2. This allows us to easily show the satisfiability of the distance constraint for D when demands are as specified in Definition 9.

The detailed proof is continued in Section E of the Appendix. There we also show $z(D) = \frac{n_1}{k_2} z(D_1) z(D_2) = q r_1 r_2$. ◀

Proof of Theorem 10. It follows from just dividing the lower bound on the network coding rate of G and the upper bound on the objective value of D obtained in Lemma 11 and Lemma 12. ◀

In the next subsection, we show how to repeatedly apply this construction. Note that, we can only apply the tensor construction on graphs in *standard form*. The following lemma allows us to tensor the new graph obtained with itself.

▶ **Lemma 13.** *Given G_1 and G_2 in standard form, Tensor(G_1, G_2, D_1, D_2) is also in standard form.*

Proof. We defer the proof of this lemma to Section E of the Appendix. ◀

3.2.4 Iterative Tensoring

In the next two statements size refers to the number of vertices in the graph A_i. The calculation of the size involves calculating the required girth at each iteration and the size of the colored bipartite graph used to tensor at each iteration.

▶ **Theorem 14.** *Given a graph $A = (V, E)$ with gap $(1+\varepsilon)$, we can construct a sequence of graphs $A_i = (V_i, E_i)$ with gap at least $(1+\frac{\varepsilon}{2})^{2^i}$, size at most $(3c_m)^{(4c_1)^{2^{i+1}}}$ where c_m and c_1 are absolute constants.*

Proof. We defer the proof to Section F of the Appendix. Let $\alpha = \frac{1+\varepsilon}{1+\varepsilon/2} - 1$. The proof first considers the α-Extension of A to start the recursion with a graph in standard form, then recursively defines pairs of tensored graphs and duals (A_i, D_i) such that the gap increases geometrically. ◀

Proof of Theorem 1. Now, we calculate an expression for the gap in terms of size. $\frac{\log(\text{gap})}{\log(1+\varepsilon/2)} \geq 2^i \geq \frac{\log\log(\text{size}) - \log\log 3c_m}{\log(16c_1^2)}$. Thus, we get a sequence of graphs with gap at least $\Omega((\log(\text{size}))^{c_2})$ where c_2 is an absolute positive constant less than 1 equal to $\frac{\log(1+\varepsilon/2)}{\log(16c_1^2)}$. ◀

4 Limits of the Construction

In this section, we show that the construction we present can not be used "as is" to prove the Li and Li conjecture. The requirement for the underlying bipartite graph to have a high girth seems to heavily contribute to the size of the graph in the next iteration. Can we do better in terms of size to yield a gap of $\omega(\log|G|)$ by choosing a smaller bipartite graph at every iteration while still having a clever upper bound on the multicommodity flow in the new graph? The answer is no. Theorem 15 states that for every colored bipartite graph B, the tensor of G_1 and G_2 with B as basis has sparsity of at least the product of the sparsities of G_1 and G_2 when the demands are all 1 in all the graphs. With the appropriate demands, this means that the sparsity grows exactly like the coding rate as in Lemma 11. Thus, for any iterative tensoring procedure that starts with a graph G with NC/MCF gap and repeatedly tensors the graph at the ith iteration (G_i) with itself or with G based on a colored bipartite graph B_i will end up with a graph G' with $\omega(\log|G'|)$ gap. Hence we can start with a graph H with a gap between the flow rate and the sparsity and apply this procedure to get a graph H' with $\omega(\log|H'|)$ gap between the flow rate and the sparsity, contradicting the bounds from [10]. This means that through iterative tensoring, if we were able to prove the conjecture, we would also prove the statement that there exists no graphs with sparsity-multicommodity flow rate gap which is clearly false.

▶ **Theorem 15.** *For any G_1, G_2, B for which $G = T(G_1', G_2', B)$ is defined (G_1' and G_2' are directed graphs obtained from G_1 and G_2 by directing each edge arbitrary in two directions such that new capacities add up to the previous),*

$$\textit{Sparsity}(G) \geq \textit{Sparsity}(G_1) \cdot \textit{Sparsity}(G_2).$$

when the demands of G_1, G_2 and G are all scaled to 1.

Proof. We defer the proof to Section G in the Appendix. ◀

Note that this gives an identical theorem as Theorem 10 for sparsity vs multicommodity flow rate for the corresponding demands.

References

1 M. Adler, N. J. A. Harvey, K. Jain, R. Kleinberg, and A. Rasala Lehman. On the capacity of information networks. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*, SODA '06, pages 241–250, Philadelphia, PA, USA, 2006. Society for Industrial and Applied Mathematics.

2 Rudolf Ahlswede, Ning Cai, S-YR Li, and Raymond W Yeung. Network information flow. *IEEE Transactions on information theory*, 46(4):1204–1216, 2000.

3 R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

4 Anna Blasiak, Robert Kleinberg, and Eyal Lubetzky. Lexicographic products and the power of non-linear network coding. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 609–618. IEEE, 2011.

5 C. Fragouli and E. Soljanin. Network coding applications. *Found. Trends Netw.*, 2(2):135–269, January 2007.

6 Z. Furedi, F. Lazebnik, A. Seress, V. A. Ustimenko, and A. J Woldar. Graphs of prescribed girth and bi-degree. *Journal of Combinatorial Theory, Series B*, 64(2):228–239, 1995.

7 N. J. A. Harvey, R. Kleinberg, and A. Rasala Lehman. Comparing network coding with multicommodity flow for the k-pairs communication problem, 2004.

8 K. Jain, V. V. Vazirani, and G. Yuval. On the capacity of multiple unicast sessions in undirected graphs. *IEEE/ACM Trans. Netw.*, 14(SI):2805–2809, June 2006.

9 G. Kramer and S. A. Savari. Edge-cut bounds on network coding rates. *J. Netw. Syst. Manage.*, 14(1):49–67, March 2006.

10 T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, November 1999.

11 Z. Li and B. Li. Network coding in undirected networks, 2004.

12 Z. Li, B. Li, D. Jiang, and L. C. Lau. On achieving optimal throughput with network coding. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, 13-17 March 2005, Miami, FL, USA*, pages 2184–2194, 2005.

13 M. Medard and A. Sprintson. *Network Coding: Fundamentals and Applications*. Academic Press. Elsevier, 2012.

14 R. W. Yeung. *Information theory and network coding*. Springer Science & Business Media, 2008.

15 R. W. Yeung, S.-Y. R. Li, N. Cai, and Z. Zhang. Network coding theory: Single sources. *Commun. Inf. Theory*, 2(4):241–329, September 2005.

A Definition of Network Coding

Let $M(i)$ be the set of all messages s_i wants to send, and let $M = \Pi_i M(i)$. For every $v \in V$, let $\text{In}(v) \subseteq E$ denote the set of edges incident to e.

▶ **Definition 16.** A *network coding solution* for a graph G specifies for each directed edge $e \in E$ an alphabet $\Gamma(e)$ and a function $f_e : M \to \Gamma(e)$ mapping the symbol transmitted on edge e. This must satisfy the following two conditions:

- Correctness: each sink node receives the message from its corresponding source, i.e. $f_{T(i)} = f_{S(i)}$.
- Causality: every message transmitted on edge e is computable from information received at its tail vertex at a time prior to the message's transmission.

▶ **Definition 17.** A *causal computation* of a network consists of

- A sequence of edges $e_1, ..., e_T$ where each edge can appear multiple times,
- A sequence of alphabets $\Lambda_1, ..., \Lambda_T$, and
- A sequence of coding functions $\rho_1, ..., \rho_T$, which in turn satisfy:
  1. For each function ρ_t such that $e_t = (u, v)$ is not a source edge, the value of ρ_t is uniquely determined by the values of the functions in the set $\{\rho_x : x < t, e_x \in \text{In}(u)\}$.
  2. For each edge e, the Cartesian product of the alphabets in the set $\{\Lambda_i : e_i = e\}$ is equal to $\Gamma(e)$.
  3. For each edge e, the set of coding functions $\{\rho_i : e_i = e\}$ together define the coding function f_e specified by the network coding solution.

At this point we are equipped with the tools needed to define the network coding rate, the information-theoretic equivalent of the flow rate.

▶ **Definition 18.** A *network coding solution* for a graph G achieves a rate r if there exists a constant $b \geq 0$ such that

- $H(S(i)) \geq r \cdot d_i \cdot b$ for each commodity i,
- for each edge $e \in E$, $H(\vec{e}) + H(\overleftarrow{e}) \leq c(e) \cdot b$,

where by $H(\vec{e})$ we denote the entropy of edge $\vec{e}$. The coding rate is defined to be the supremum of the rates of all network coding solutions.

B Multicommodity Flows

The standard LP formulation for concurrent multicommodity flow problems is written below. It has a variable for every path $p \in P_i$, where P_i is the set of all paths between s_i and t_i. We want to find the largest rate λ that can be concurrently sent between all source-sink pairs subject to the path variables being non-negative and not exceeding the capacity of any edge over all commodities.

$$\begin{aligned} \text{maximize} \quad & \lambda \\ \text{subject to} \quad & \sum_{p \in P_i} f(p) \geq \lambda d_i \quad \forall i \in [k] \\ & \sum_{p : e \in p} f(p) \leq c(e) \quad \forall e \in E \\ & f(p) \geq 0 \qquad \forall p \\ & \lambda \geq 0 \end{aligned} \tag{5}$$

Proof of Lemma 4. We contract all the edges with zero weight in the dual. We need to show that the gap does not decrease. Removing a zero dual variable from a multicommodity solution cannot improve the flow rate, since the distances and the dual objective remains the same. We can use the same coding solution for the new graph with the exception that we now compose the encoding on the edges that were contracted. This shows that the flow rate does not increase and the coding rate does not decrease, proving that their ratio does not decrease. ◀

C Standard Form

This section gives the detailed description of G_α and $D_\alpha(G)$. Let k_v be the number of sources and sinks at vertex $v \in V(G)$. In the graph G_α, add k_v edges (leaves) at v with capacity $z(D)d(1+\varepsilon)$ and shift all the sources or sinks at v to the unique endpoints of these leaves. As each source sends $\geq z(D)d(1+\varepsilon)$ amount of information in an optimal network coding solution and can still send $z(D)d(1+\varepsilon)$, the network coding rate doesn't decrease below $z(D)(1+\varepsilon)$. We construct $D_\alpha(G)$ as follows:

1. For each edge originally in G, assign the same weights as in D.
2. Give weight $\frac{\alpha}{kd(1+\varepsilon)}$ to the new edges.

Distances in the dual don't decrease, so $D_\alpha(G)$ is a valid solution. Since we added k new edges, $z(D_\alpha(G)) = kz(D)d(1+\varepsilon)\frac{\alpha}{kd(1+\varepsilon)} + z(D) = z(D)(1+\alpha)$. Thus, $\frac{NC_G}{z(D\alpha(G))} \geq \frac{1+\varepsilon}{1+\alpha}$.

D Colored Bipartite Graph Construction

In this section, we give a construction for a colored bipartite graph C_B in $\mathcal{B}_{n_1,n_2,r,s,2g,r,s}$ $\forall r, s, g \geq 3$ with $n_1, n_2 \leq rs(9rs)^{g+2}$. We start with a (r, s)-biregular bipartite graph with girth at least $2g$.

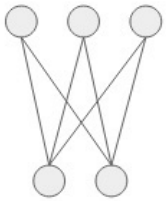


Figure 4 (2,3) biregular bipartite graph with girth 4.

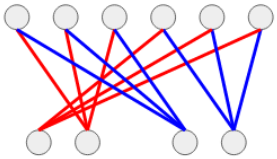


Figure 5 Intermediate graph when one set of colors hasf been assigned.

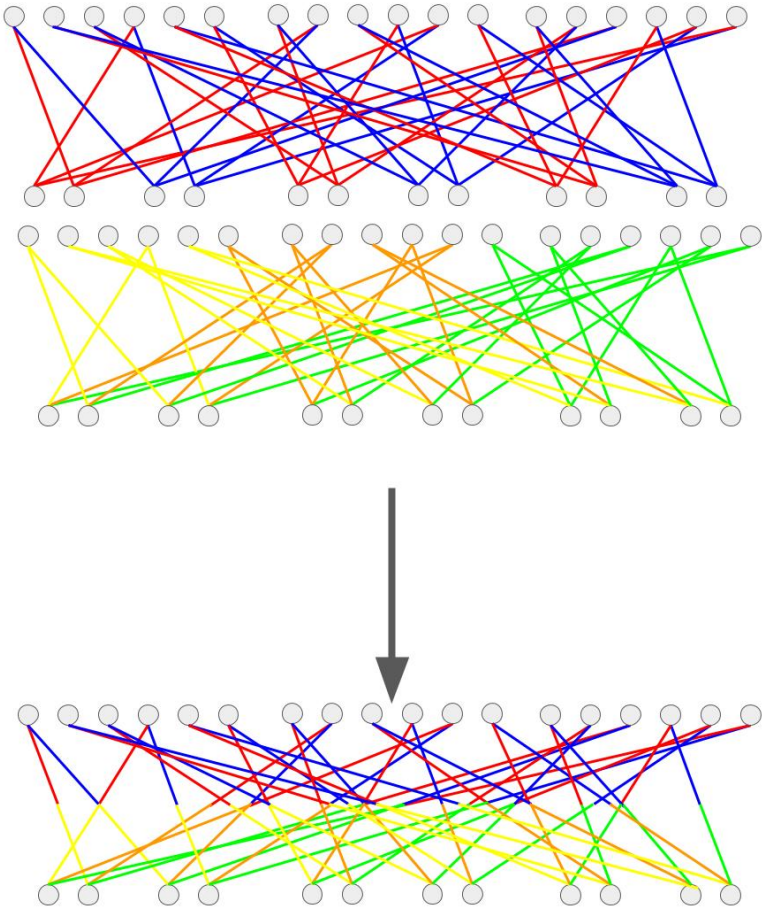


Figure 6 Final colored bipartite graph with girth 4.

▶ **Lemma 19.** *[6] For all $r, s, g \geq 3$, there exists a (r, s)-biregular bipartite graph with girth at least $2g$ and having at most $n = (9rs)^{g+2}$ vertices.*

This lemma follows from Theorem E in [6].

Proof of Lemma 8. Let $B(r, s, g)$ be a graph satisfying the above property. For simplicity, denote $B(r, s, g)$ by just $B = (V_1, V_2, E)$. Denote the coloring for every edge e by (a_e, b_e). First we construct an intermediate graph H in $\mathcal{B}_{n'_1,n'_2,r,s,2g,1,s}$ $(n'_1 = s|V_1|, n'_2 = s|V_2|)$ as follows:

1. Enumerate all the edges incident to a vertex $v \in V_2$ as $e_v^{(1)}, ..., e_v^{(s)}$.
2. Add s copies of V to graph H. Enumerate these copies as $(V_1^{(1)}, V_2^{(1)}), ..., (V_1^{(s)}, V_2^{(s)})$.
3. $\forall v^j \in V_2^{(j)}, j \in [s]$, $\forall i \in [s]$, corresponding to edge $e_v^{(i)} = (u, v) \in E(B)$, add an edge e from v^j to $u^{(((j+i-2) \bmod s)+1)} \in V_1^{(((j+i-2) \bmod s)+1)}$ (copy of u in $(((j + i - 2) \bmod s) + 1)$-th copy of V_1). Set $b_e = ((j + i - 2) \bmod s) + 1$. Therefore, $\forall u^j \in V_1^{(j)}$, edges e' incident at u^j have $b_{e'} = j$ (same color).

For a vertex $v \in V_2^{(j)}$, the edge corresponding to $e_v^{(i)}$ comes from a vertex in $V_1^{(((j+i-2) \bmod s)+1)}$. Thus, all edges incident to v have distinct colors.

We still need to show that the girth of H is at least $2g$. For this, we show that a cycle C of length c in H implies a cycle of length $\leq c$ in B. As all the edges incident to a vertex in H correspond to different edges in B, when we project back C to a cycle C' in B, no two consecutive edges in C' are the same implying C' has no cycle of length 2. Thus, C' must have a cycle of length $3 \leq c' \leq c$. B has girth at least $2g$, so the girth of H cannot be smaller.

Now, we repeat the process for $H = (H_1, H_2)$ to get graph C_B with H_1 playing the role of V_2 and H_2 playing the role of V_1 in the above algorithm. This time we assign $a_e \in [r]$ and make r copies of H. We can see that as was the case for b_e, each vertex in a copy of H_1 gets r distinct a_e values and each vertex in a copy of H_2 gets the same a_e depending on which copy it belongs to. The girth doesn't decrease on going from H to C_B giving us the result we claim. ◀

An example of a colored bipartite graph in $\mathcal{B}_{12,18,3,2,4,3,2}$ is given in Figure 6. We start with $K_{2,3}$ as in Figure 4 with girth 4. Then, we construct the intermediate graph in $\mathcal{B}_{4,6,3,2,4,1,2}$ as shown in Figure 5. The color of the edge depends on the copy it is incident to on the lower side. For a vertex on the upper side, we send edges to correct vertices in distinct copies cyclically.

E Gap Amplification Proofs

Proof of Lemma 11. Graph G_1 has a network coding rate of at least $r_1(1 + \varepsilon_1)$ and hence each source sends $r_1(1 + \varepsilon_1)d_1$ amount of information to its corresponding sink, and similarly for G_2. This is true even for the directed graphs G'_1 and G'_2 by definition. While constructing $T(G'_1, G'_2, B')$, we aligned the source-sink pair in the same direction as the directed edge. This allows us to compose the network coding solutions (N_2 over N_1) to get the information sent from each source in G to be at least $r_1r_2(1+\varepsilon_1)(1+\varepsilon_2)d_1d_2$. This is due to the fact that as we are replacing edges in G'_1 by a source-sink pair of a copy of G'_2, the effective capacity seen by the replaced edge (e) with capacity c_{1e} is now $c_{1e}r_2(1 + \varepsilon_2)d_2$. Thus, the coding rate for graph G is at least $\frac{r_1r_2(1+\varepsilon_1)(1+\varepsilon_2)d_1d_2}{(\text{demand in graph } G)} = \frac{r_1r_2(1+\varepsilon_1)(1+\varepsilon_2)d_1d_2q}{d_1d_2} = r_1r_2(1+\varepsilon_1)(1+\varepsilon_2)q.$ ◀

Proof of Lemma 12. Here, we prove that D is indeed a valid dual solution. B' has $n_1 = k_2q$ nodes on the left side. Let $l_1(s_i, t_i)$ denote the shortest distance between i-th source-sink pair

with respect to dual D_1. Let $l_2(s_i, t_i)$ denote the shortest distance between i-th source-sink pair with respect to dual D_2.

Let $G_1'^u$ and $G_2'^u$ be the undirected version of the graphs G_1' and G_2' respectively. $G_1'^u$ and $G_2'^u$ are graphs G_1 and G_2 where each edge is divided into 2 edges with capacities adding up to the previous one. Construct dual solutions D_1' and D_2' for $G_1'^u$ and $G_2'^u$ such that each divided edge still gets the same weight as in dual solutions D_1 and D_2. The distances between source-sink pairs remain the same. In G, calculate the shortest distance, i.e. $l(s_i^{(y)}, t_i^{(y)})$ between source-sink pair $(s_i^{(y)}, t_i^{(y)})$ which corresponds to the i-th source-sink pair (s_i, t_i) in the y-th copy of $G_1'^u$ (finally, we make the graph undirected). In this copy of $G_1'^u$, assume that we replaced each edge with the j_y-th source-sink pair of $G_2'^u$ (this is unique due to Property (2) in Definition 6). Therefore, according to D, $l(s_i^{(y)}, t_i^{(y)}) \leq l_1(s_i, t_i) l_2(s_{j_y}, t_{j_y})$ (these correspond to dotted paths). Any other path from $s_i^{(y)}$ to $t_i^{(y)}$ involves traversing to another copy of $G_1'^u$ through a copy of $G_2'^u$ that replaced edges in this copy of $G_1'^u$. This transition from a copy of $G_1'^u$ to another copy of $G_1'^u$ in G corresponds to two consecutive edges in the bipartite graph B'. Any such path in G having no loops would thus have to make at least g of these transitions to revert back to the original copy of $G_1'^u$ containing the source. Here, the girth of graph B' is at least $2g$. Each transition involves crossing at least one edge (in a copy of G_2') with weight at least $w_1 w_2$ in D because G_2, being in standard form, has all sources and sinks lying on distinct vertices and vertices of a copy of G_1' connect only to the vertices of a copy of G_2' carrying a unique source or sink. Thus, such a path would have distance at least $g w_1 w_2 = \frac{l_2 l_1}{w_1 w_2} w_1 w_2 \geq l_1 l_2$ using l_1, l_2, w_1, w_2 from Definition 9. The cheating paths have distance at least $l_1 l_2$ implying $l(s_i^{(y)}, t_i^{(y)}) = l_1(s_i, t_i) l_2(s_{j_y}, t_{j_y})$. The left hand side of the distance constraint in LP 4 becomes $\sum_{i=1,y=1}^{k_1,n_1} \frac{d_1 d_2}{q} l(s_i^{(y)}, t_i^{(y)}) \geq 1$, where the first expression in the summand is the demand of source-sink pairs in G. Now, we are going to show that the constraint is true:

$$\sum_{i=1,y=1}^{k_1,n_1} \frac{d_1 d_2}{q} l(s_i^{(y)}, t_i^{(y)}) = \sum_{i=1,y=1}^{k_1,n_1} \frac{d_1 d_2}{q} l_1(s_i, t_i) l_2(s_{j_y}, t_{j_y}) = \frac{d_1 d_2}{q} \sum_{y=1}^{n_1} l_2(s_{j_y}, t_{j_y}) \sum_{i=1}^{k_1} l_1(s_i, t_i)$$

$$= \frac{1}{q} \cdot \frac{n_1}{k_2} \left(\sum_{j=1}^{k_2} d_2 l_2(s_j, t_j) \right) \left(\sum_{i=1}^{k_1} d_1 l_1(s_i, t_i) \right) \geq \frac{n_1}{q k_2} = 1$$

The second to last equality follows from the fact that there are total n_1 copies of G_1', j_y is fixed for fixed y-th copy of G_1' and each $l_2(s_j, t_j)$ ($j \in [k_2]$) is thus counted $\frac{n_1}{k_2}$ time. The last inequality follows from D_1' and D_2' being valid dual solutions of $G_1'^u$ and $G_2'^u$ respectively (distance constraints). This proves that D is a valid dual solution.

The value of $z(D_1')$ for graph $G_1'^u$ is r_1. D_1' assigns the same dual weights as that of D_1 for the divided edges and is a valid dual solution for G_1', and similarly for D_2'. We can see from the construction of D and the edge capacities that $z(D) = \frac{n_1}{k_2} z(D_1') z(D_2') = q r_1 r_2$. D is a function of G_1, G_2, D_1, D_2. ◀

Proof of Lemma 13. Demands are equal for all source-sink pairs in $G = \text{Tensor}(G_1, G_2, D_1, D_2)$ by definition. We need to prove that all sources and sinks in G still lie on distinct vertices. We don't add any new source-sink pairs and thus, each source-sink pair lies on distinct vertices on a copy of G_1'. While constructing $T(G_1', G_2', B')$, we merge a vertex v in a copy of G_1' with a source or a sink vertex of a copy of G_2' and since each vertex contains a unique source or sink of G_2', no two vertices from different copies of G_1' are merged together. This implies that all sources and sinks still lie on distinct vertices of G. ◀

F Proof of Theorem 14

Proof. Using Lemma 4, we can assume that graph A has an optimal dual solution D with all dual variables being non-zero. It is without loss of generality that A has equal demands for all source-sink pairs. Define A^* to be the α-Extension of A given D and $D^* = D_\alpha(A)$ $(1+\alpha = \frac{1+\varepsilon}{1+\varepsilon/2})$. Let A^* have c_n vertices, c_m edges and c_k source-sink pairs having c_d demand each. Without loss of generality we can assume that $c_m \geq c_k, c_n$ as otherwise we can just divide some edges into multiple edges with reduced capacities. Let l be the largest distance between any source-sink pair in the dual D^* and $w > 0$ be the minimum weight of an edge in dual D^*. We also know that $\frac{NC_{A^*}}{z(D^*)} \geq \frac{1+\varepsilon}{1+\alpha} = 1 + \frac{\varepsilon}{2}$. As the objective value of any dual solution is at least the flow rate, we get that A^* has a gap of at least $(1 + \frac{\varepsilon}{2})$. A^* is in standard form. A_i is defined iteratively as follows:

$A_0 = A^*, D_0 = D^*, \varepsilon_0 = \frac{\varepsilon}{2}$.
For $i \geq 1$:
ε_i is such that $(1 + \varepsilon_i) = (1 + \varepsilon_{i-1})^2$.
A_i =Tensor$(A_{i-1}, A_{i-1}, D_{i-1}, D_{i-1})$.
D_i =Dual$(A_{i-1}, A_{i-1}, D_{i-1}, D_{i-1})$.

Note that $\forall i, A_i$ is in standard form using Lemma 13 and thus iterative tensoring is valid. Through Theorem 10, we know that if $\frac{NC_{A_{i-1}}}{z(D_{i-1})} \geq (1+\varepsilon_{i-1})$, then $\frac{NC_{A_i}}{z(D_i)} \geq (1+\varepsilon_{i-1})^2 = 1+\varepsilon_i$. As $\frac{NC_{A^*}}{z(D^*)} = 1 + \frac{\varepsilon}{2}$, we get $\frac{NC_{A_i}}{z(D_i)} \geq 1 + \varepsilon_i = (1 + \varepsilon/2)^{2^i} \forall i$ by induction. The objective value of any dual solution is at least the flow rate implying that the gap between coding and flow rate for A_i is at least $(1 + \frac{\varepsilon}{2})^{2^i}$.

To see how the size of A_i grows, we first calculate the required girth $(2g_i)$ at each iteration. From the construction of $D_i = \text{Dual}(A_{i-1}, A_{i-1}, D_{i-1}, D_{i-1})$ in the proof of Lemma 12 we see that $w_i = w_{i-1}^2, l_i \leq l_{i-1}^2$. By induction, we have that for all i, $w_i = w^{2^i}$ and $l_i \leq l^{2^i}$. From Definition 9, we have that $g_i = \frac{l_{i-1}^2}{w_{i-1}^2} \leq \frac{(l^{2^{i-1}})^2}{(w^{2^{i-1}})^2} = (\frac{l}{w})^{2^i}$. Therefore, $g_i \leq (\frac{l}{w})^{2^i} \forall i \geq 1$. Let $c = \frac{l}{w} \geq 1$.

Now, we establish an upper bound on the size of the graph. Recall A_i is the $T(A'_{i-1}, A'_{i-1}, B_i)$ where $B_i = C_{m'_{i-1} k_{i-1} g_i}$ and $m'_{i-1} = 2m_{i-1}$. A'_{i-1} is the directed graph constructed according to the optimal network coding solution of A_{i-1}. Let $n_{1i} = |V_i(B_i)|, n_{2i} = |V_2(B_i)|$. From Lemma 8, $n_{1i} \leq (9m_{i-1}k_{i-1})^{g_i+3} \leq (9m_{i-1}k_{i-1})^{c^{2^i}+3}$.

Note that $m_i = \frac{n_{1i}}{k_{i-1}} m_{A'_{i-1}} m_{A'_{i-1}} = \frac{n_{1i}}{k_{i-1}} (4m_{i-1}^2)$ and $k_i = n_{1i}k_{i-1}$. Each edge in A'_{i-1} is replaced by a copy of A'_{i-1} and each copy is counted k_{i-1} times implying $v_i \leq 2m_{i-1}v_{i-1}\frac{n_{1i}}{k_{i-1}}$.

Moreover, $\frac{m_i}{k_i} = 4(\frac{m_{i-1}}{k_{i-1}})^2$. By induction, $k_i \leq m_i$ as $c_k \leq c_m$. Likewise, we get that $\frac{m_i}{v_i} = 2\frac{m_{i-1}}{v_{i-1}} \geq 1 \forall i$. The upper bound on m_i (the number of edges in A_i) is as follows:

$m_i \leq 4n_{1i}(m_{i-1}^2) \leq 4m_{i-1}^2(9m_{i-1}k_{i-1})^{g_i+3} \leq (9m_{i-1}^2)^{g_i+4} = (3m_{i-1})^{2c^{2^i}+8} \leq (3m_{i-1})^{2(c+1)^{2^i}+8} \leq (3m_{i-1})^{4(c+1)^{2^i}} \forall i \geq 1$ $(c \geq 1)$.

Let $c_1 = c + 1$.

▶ **Claim 20.** $m_i \leq (3c_m)^{(4c_1)^{2^{i+1}}}$.

Proof. For $i = 0$, the right hand side evaluates to $(3c_m)^{(4c_1)^2} \geq c_m$, which is equal to the left hand side. Now we assume that the statement is true for $i - 1$ and prove for i where $i \geq 1$. $m_i \leq (3m_{i-1})^{4c_1^{2^i}} \leq (3(3c_m)^{(4c_1)^{2^i}})^{4c_1^{2^i}} = 3^{(4c_1)^{2^i} 4c_1^{2^i} + 4c_1^{2^i}} c_m^{(4c_1)^{2^i} 4c_1^{2^i}} \leq (3c_m)^{(4c_1)^{2^{i+1}}}$ as $4^{2^i+1} + 4 \leq 4^{2^{i+1}} \forall i \geq 1$.

We have $v_i \leq m_i$. Thus, the size of graph A_i is at most $(3c_m)^{(4c_1)^{2^{i+1}}}$. ◀

◀

G Proof of Theorem 15

Proof. Think of G_1 and G_2 as undirected G_1' and G_2'; their sparsity remains the same. Let H be the set of edges on the cut that achieves the sparsest cut on G separating n source-sink pairs. Consider partitioning this set into sets $H_i = \{e_{1i}, e_{2i}, ..., e_{h_i i}\}$ according to which copy of G_2 (or equivalently G_2'), the edge belongs to in G. H_i denotes the edges belonging to the i-th copy of G_2, $|H_i| = h_i$. Note that $|H| = \sum_i h_i$. Let $n_i^{(2)}$ be the number of source and sink pairs that H_i separates in the i-th copy of G_2. These cuts have capacity $\sum_{e\in H_i} c_{2e}$ in G_2. By construction, each of these source-sink pairs would have replaced an edge in some copy of G_1 (or equivalently undirected G_1'). Assume the k-th ($k \in [n_i^{(2)}]$) source-sink pair replaced edge e_i in the j_{ik}-th copy of G_1 (All source-sink pairs replace the same edge). Mark this edge in the j_{ik}-th copy of G_1 (which has now been replaced in G). The i-th copy of G_2 makes $n_i^{(2)}$ marks. Let F_j be the set of all such marked edges in the j-th copy of G_1. Let F_j cut $n_j^{(1)}$ source-sink pairs in G_1. Any source-sink pair that gets cut in G by H must be cut in G_1 under F_j by construction. Therefore, $\sum_j n_j^{(1)} \geq n$. It is not an equality because there could be a source-sink pair that gets cut by F_j but not by H in G, due to paths that travel from the source to other copies of G_1 through connecting copies of G_2 and come back at the sink. The theorem follows from the following inequalities:

$$\begin{aligned}
\sum_{e\in H} c_e &= \sum_i c_{1e_i} \sum_{e\in H_i} c_{2e} = \sum_i n_i^{(2)} c_{1e_i} \frac{\sum_{e\in H_i} c_{2e}}{n_i^{(2)}} \\
&\geq \sum_i n_i^{(2)} c_{1e_i} \text{Sparsity}(G_2) = \text{Sparsity}(G_2) \left(\sum_i n_i^{(2)} c_{1e_i} \right) \\
&= \text{Sparsity}(G_2) \left(\sum_j \sum_{e\in F_j} c_{1e} \right) = \text{Sparsity}(G_2) \left(\sum_j n_j^{(1)} \frac{\sum_{e\in F_j} c_{1e}}{n_j^{(1)}} \right) \\
&\geq \text{Sparsity}(G_2) \left(\sum_j n_j^{(1)} \right) \text{Sparsity}(G_1) \geq n \left(\text{Sparsity}(G_1) \cdot \text{Sparsity}(G_2) \right)
\end{aligned} \tag{6}$$

The first equality follows from the definition of edge capacities in G in terms of edge capacities in G_1 and G_2. Since H_i cuts $n_i^{(2)}$ source-sink pairs in a copy of G_2, the first inequality follows from the Sparsity(G_2) being the smallest ratio for all the cuts. The first equality on the third line follows from the fact that an edge belongs to F_j only when the corresponding source-sink pair that replaced this edge in G_1 is cut by the cut corresponding to that copy of G_2 and i-th copy of G_2 result in exactly $n_i^{(2)}$ such edges distributed amongst F_js. Therefore, $\frac{\sum_{e\in H} c_e}{n} \geq \text{Sparsity}(G_1) \cdot \text{Sparsity}(G_2) \implies \text{Sparsity}(G) \geq \text{Sparsity}(G_1) \cdot \text{Sparsity}(G_2)$. Here, we assumed that all the demands are 1 in graphs G_1, G_2 and G. ◀

Compression in a Distributed Setting

Badih Ghazi[*1], **Elad Haramaty**[†2], **Pritish Kamath**[‡3], **and Madhu Sudan**[§4]

1 **Massachusetts Institute of Technology, Cambridge, USA**
badih@mit.edu
2 **Harvard University, Cambridge, USA**
seladh@gmail.com
3 **Massachusetts Institute of Technology, Cambridge, USA**
pritish@mit.edu
4 **Harvard University, Cambridge, USA**
madhu@cs.harvard.edu

Abstract

Motivated by an attempt to understand the formation and development of (human) language, we introduce a "distributed compression" problem. In our problem a sequence of pairs of players from a set of K players are chosen and tasked to communicate messages drawn from an unknown distribution Q. Arguably languages are created and evolve to compress frequently occurring messages, and we focus on this aspect. The only knowledge that players have about the distribution Q is from previously drawn samples, but these samples differ from player to player. The only *common* knowledge between the players is restricted to a common prior distribution P and some constant number of bits of information (such as a learning algorithm). Letting T_ϵ denote the number of iterations it would take for a typical player to obtain an ϵ-approximation to Q in total variation distance, we ask whether T_ϵ iterations suffice to compress the messages down roughly to their entropy and give a partial positive answer.

We show that a natural uniform algorithm can compress the communication down to an average cost per message of $O(H(Q) + \log(D(P||Q)))$ in $\tilde{O}(T_\epsilon)$ iterations while allowing for $O(\epsilon)$-error, where $D(\cdot||\cdot)$ denotes the KL-divergence between distributions. For large divergences this compares favorably with the static algorithm that ignores all samples and compresses down to $H(Q) + D(P||Q)$ bits, while not requiring $T_\epsilon \cdot K$ iterations that it would take players to develop optimal but separate compressions for each pair of players. Along the way we introduce a "data-structural" view of the task of communicating with a natural language and show that our natural algorithm can also be implemented by an efficient data structure, whose storage is comparable to the storage requirements of Q and whose query complexity is comparable to the lengths of the message to be compressed. Our results give a plausible mathematical analogy to the mechanisms by which human languages get created and evolve, and in particular highlights the possibility of coordination towards a joint task (agreeing on a language) while engaging in distributed learning.

1998 ACM Subject Classification E.4 Coding and Information Theory

Keywords and phrases Distributed Compression, Communication, Language Evolution, Isolating Hash Families

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.19

[*] The author is supported in part by NSF STC Award CCF 0939370 and NSF Award CCF-1217423.
[†] The author is supported in part by NSF Award CCF-1565641.
[‡] The author is supported in part by NSF CCF-1420956.
[§] The author is supported in part by NSF Award CCF-1565641 and a Simons Investigator Award.

© Badih Ghazi, Elad Haramaty, Pritish Kamath and Madhu Sudan;
licensed under Creative Commons License CC-BY

1 Introduction

Motivated by the goal of understanding human communication and in particular phenomena associated with the formation and development of language, we introduce a distributed compression problem and study it. We start with a description of the compression problem first, and then give our motivation.

1.1 Model

The Basic Model

We consider a distributed setting where K players, with a complete network of point-to-point connections, are exchanging a sequence of messages drawn from an, apriori unknown, distribution Q. In our model the set of possible messages is a countable set, and we use $\mathbb{N}$, the set of natural numbers to denote this set without loss of generality. The communication proceeds in rounds: In round t, a message m is chosen from $\mathbb{N}$ according to Q independent of the past. Simultaneously an ordered pair of players $i, j \in [K] \stackrel{\text{def}}{=} \{1, \ldots, K\}$ with $i \neq j$ is chosen uniformly from all such pairs. The goal is for player i to encode the message m into a sequence of bits and send it to player j. Player j receives this sequence of bits and decodes it to a message $\hat{m}$. (Note that the encoding, and decoding, may depend on the history of interactions involving the sender, respectively receiver.) The round t is said to have an error if $m \neq \hat{m}$. The goal is to design encoding and decoding schemes that satisfy the condition that for every round t, the probability of error, over the history of random choices, is at most ε and the measure of performance is the expected length of communication averaged over the rounds up to t, studied as a function of t.

Efficiency Issues

A second measure of performance of the encoding and decoding algorithms is their "computational efficiency". We define this notion using a "data-structural" perspective. Note that any encoder or decoder essentially needs to learn and store (approximations to) the distribution Q in order to perform moderately well. Thus, such an encoder or decoder needs to work with the amount of space that it might take to remember Q. At the same time, encoding or decoding a single message should not, and need not, take time linear in the storage. We thus measure the efficiency of the encoding and decoding algorithms in terms of its *space* requirement, and its *processing time* to compute the encoding of a message m including the time it takes to update its memory to incorporate this new message in its history.

Setup Assumptions

Finally, we parameterize one commonality in the initialization of the different players. Note that to initialize any communication the players must have some way of exchanging messages. One may consider the natural binary description of messages as one such possibility. Other possibilities may go via Kolmogorov complexity, i.e., by letting the players share a common universal machine and then representing a message via the encoding of machine that outputs the binary representation of the message and halts. Rather than choosing any one of these representations, we parametrize the setup by the exact initial representation. More precisely, we consider a distribution P on $\mathbb{N}$ for which a given initial representation is optimal and

assume that all players share P at the outset[1]. Thus the encoding and decoding algorithms may depend on this prior distribution, but otherwise the algorithms must be completely uniform and may not rely on any other shared information. Note that Q is chosen adversarially and has no relationship to P, however we will allow our performance, i.e., the expected length of the compression to depend on the gap (distance or divergence) between P and Q.

1.2 Motivation: Language Formation and Development

Our motivation to study this distributed compression problem is to give a fresh perspective on phenomena associated with the formation and evolution of human languages. We note that the study of languages is a central quest in linguistics, cognitive science and philosophy and much is known about it based on empirical studies. Our hope is to add some mathematical flavors to this.

For our purpose, we may view language as providing a map that describes how to convert a message in an individual's brain into a sequence of utterances. Yet no language has a short description of this map. Part of the challenge seems to be that language is constantly evolving and if one were to fix any bound, language seems to evolve to a point where the description length exceeds this bound. The reason for this evolution may be viewed as some form of compression. While the ultimate goal may not be the time it takes to convey a message, language certainly evolves by creating shortcuts for currently frequent messages[2]. This motivates our use of the compression capability of the message-to-utterance map as a crude measure of performance. It is not the unique goal, but it is well-aligned with the goals of language.

A second feature about languages is that no two individuals probably have identical descriptions of the map. Attempts to give a unified description of the language (say, as in a dictionary) end up with many different dictionaries and each one capturing some segment of the population. Yet, language is robust to this variation and for the most part, communication manages to work despite the lack of agreement on the dictionary. Our view of this diversity is to consider the process of language acquisition. Individuals (children) learn from examples and indeed there is major diversity in the set of examples one encounters depending on one's own circumstances, but even if one were to factor out this diversity (e.g., by considering identical twins), their experiences are still different. This inspires our setting: individuals are all born identical and get samples from the same distribution. (Furthermore there are no network effects - the underlying graph is a complete graph and the message distribution is independent of the edge distribution. We will discuss this shortly.) Yet their samples are not identical and even this minor discrepancy seems to foil simple algorithms to coordinate on a compression map and introduces either diversity in the map, or complexity in the coordination process. Thus, the distributed compression problem already gives a potential reason for the diversity in language.

We emphasize that our choice of a simple graph (the complete graph) and the independence between the messages and the graph are not restrictions of the "model". It is quite easy to extend our model to the setting where the graphs are complex, the distributions on edges are weighted and to allow the distribution of the messages to depend on the edge. While

[1] Intuitively, we can think of P as being a primitive "gesturing language" that is understandable to all people.

[2] For instance, a language could evolve to use the word "Ix" to denote "a boy who is not able to satisfactorily explain what a Hrung is, nor why it should choose to collapse on Betelgeuse Seven" [1].

such richness is permitted by the model, we restrict to the simple setting to allow simpler contrasts between basic options (and our more sophisticated one).

Finally, one intriguing aspect of language is the amount of influence that different players have on its development. For the most part, language evolution seems to be a decentralized process, but this does not imply equal influence for all players. The role of books, especially those on grammar or dictionaries, of the media, and popular figures definitely assigns disproportionate influence to different players. A question that might be asked is whether language could manage to gain coherence across the population in the absence of such highly influential figures. Our model offers a way to study such questions (in our simplified setting of compression).

1.3 Context and Main Benchmarks

Our main result is a distributed compression algorithm with "decent" performance. To set the stage for this algorithm, we first describe some basic benchmarks and then some basic compression schemes.

In what follows, we use $Q(m)$ to denote the probability of a message m being drawn according to distribution Q. We let $H(Q) = \sum_{m\in\mathbb{N}} Q(m)\log_2(1/Q(m))$ denote the binary entropy of Q and we let $D(Q||P) = \sum_{m\in\mathbb{N}} Q(m)\log_2(Q(m)/P(m))$ denote the KL-divergence between Q and P. The best possible compression scheme would need at least $H(Q)$ bits per message in expectation – this is true in the 2-person case and we will discuss below whether this is achievable in the distributed setting.

We refer to $\tau = 2t/K$ as the *local time*, which roughly measures the number of messages any one player has seen (either as sender or receiver) at time t. We use T_ε to denote the local time by when a fixed player can obtain a ε-close approximation to Q, with probability at least $1-\varepsilon$. Note that T_ε can be upper bounded by $O(2^{H(Q)/\varepsilon})$ (and so in particular T_ε is finite for distributions with finite entropy). Intuitively, T_ε is a reasonable measure of local time by which one may expect to be able to compress well according to Q (even in the simple 2-player setting) and this will be a benchmark time for our compression algorithms also.

Finally, a natural upper bound on the space complexity of storing (an ε-approximation to) Q is again $2^{H(Q)/\varepsilon}$. We will compare the storage needs of the various solutions below to this benchmark. Natural measures of update times would be polylogarithmic in space and we will ask for that. (In what follows, we assume messages are given as black boxes that can be stored in unit time and space and that basic operations such as comparison of messages (is $m_1 \le m_2$?) take unit time.)

We now turn to some basic schemes for compression.

Near Ideal Compression: We first point out the (obvious?) flaw with the most natural hope one may have: Players could try to learn Q and get ε-close to the right distribution moderately fast (in local time T_ε) and then use the optimal (Huffman) coding applied to such a distribution. Unfortunately, they can not agree on this naive distribution and so no naive variation of the 2-player compression mechanism seems to be implementable.

Static Compression: Players simply encode and decode according to the Huffman code for distribution P. The error probability is zero and the expected length of the compression will be at most $H(Q) + D(Q||P) + O(1)$. The good news with this scheme is that the performance does not depend on K, but the bad news is that players do not learn to speak more effectively from examples. This is captured by the fact that the gap from optimal compression is $D(Q||P)$ and we think of this as a large quantity.

Point-to-Point Compression: For every ordered pair (i, j), player i uses the Lempel-Ziv (or any universal) compression algorithm restricted to the sequence of messages that were directed from i to j and player j decodes according to the same history. This scheme converges to a compression length of $H(Q)$ but it takes a relatively long time - a local time of $K \times T_\varepsilon$. We view dependence on K in the local time as too high. This scheme also involves memory requirement which is K times larger than the space needed for a 2-player solution.

Dictatorial Compression: Here one player (the dictator) is singled out and tasked with the compression task. He learns a distribution close to Q and then communicates the resulting encoding/decoding scheme to all other players. The compression achieved by this scheme is near-optimal (converges to $H(Q)$); and the space requirement is also near-optimal. The main quantitative weakness we see is a mild dependence on K in the time it takes for this scheme to converge: Specifically it takes about T_ε local time for the dictator to learn the distribution (which is perfectly fine), but then it needs to spread the information out to all K players and this takes $T_\varepsilon + \Theta(\log K)$ additional local time (using any reasonable gossip algorithm with proper pipelining of messages). The main "criticism" of the scheme may be that it is centralized. While centralized mechanisms do plausibly play a role in the development of languages, they do not seem to be the only mechanism, and so we seek a truly distributed solution below.

1.4 Results

We now state our main theorem.

▶ **Theorem 1** (Main Theorem). *Let $\varepsilon > 0$ be a sufficiently small positive absolute constant. For all K and P, there exists a deterministic distributed compression protocol Π, such that for any distribution Q over $\mathbb{N}$ when run for T iterations,*

- *the amortized communication cost of Π over T iterations approaches $O(H(Q)+\log D(Q||P)+\log(1/\epsilon))$ as T gets large. More formally, the amortized communication cost is*

$$O\left(H(Q)+\log D(Q||P)+\log(1/\epsilon)+\frac{2^{\Theta(H(Q)+D(Q||P))/\varepsilon}\cdot K}{T}\cdot D(Q||P)+1\right)$$

- *in each round, the transmitter and receiver run in time linear in their input and output sizes.*
- *the space usage is exponential in $(H(Q)+D(Q||P))/\epsilon$.*

Our scheme is obtained with each player mixing the static scheme (used initially) with a switch to a more complex scheme once a sufficiently good approximation to Q has been learned (by the player). A central ingredient in our scheme is a solution to the "Uncertain Compression" problem studied by Juba et al. [6] and Haramaty and Sudan [4]. In the uncertain compression problem, two players attempt to compress a single message drawn from a distribution Q, but only the sender knows P and the receiver only knows some distribution Q' which is close to Q. The uncertain compression problem seems to arise naturally in our setting (neither the sender nor the receiver know Q in our case, but both are close and this mild difference can simply be ignored). [6] give a "randomized" solution to this problem which compresses messages roughly down to $H(Q)+O(1)$ bits. Adapting this solution to our setting, essentially as a black box, achieves similar effects in our setting (compression down to $H(Q)+\delta\cdot D(Q||P)$ bits in time $\delta^{-1}\cdot T_\varepsilon$ local time), but a flaw with this scheme is that it requires the players to share a large random string in the setup phase.

Instead we turn to the solution of [4] which does not need any randomness, but their solution assumes that Q is supported on a finite set (of size N) and their compression length is $O(H(Q) + \log\log N)$. Since our distributions are not supported on finite sets, we need to modify their scheme and a careful modification followed by a relatively straightforward analysis leads to our eventual scheme and analysis. In the process, we are also able to build a small data structure implementing the encoding and decoding with efficient processing time. We point out that if one does not care about computational efficiency, then we can remove the additive $\log(1/\epsilon)$ term from the communication cost in Theorem 1 above while also replacing the multiplicative $2^{\Theta(H(Q)+D(Q||P))/\varepsilon}$ factor by $2^{\Theta(H(Q))/\varepsilon}$ (for more details, see Section 3.6).

1.5 Previous Work on Language Evolution

There have been many works on language evolution (to the best of our knowledge all from outside the theoretical computer science community). Without trying to be exhaustive, we briefly mention some of them. In the lingustics field, significant work has been done in the last decades on trying to understand language evolution, including [2, 3]. Several papers also study language from the landscape of evolutionary game theory and evolutionary biology, e.g., [14, 12, 13, 9, 10, 11, 5, 7, 8], and neuroscience, e.g., [16]. There has also been some previous attempts to connect language evolution to the framework of information theory (e.g., [15]), but their focus is on word formation in the "two-player" case, unlike our setup where we consider language as the outcome of the interaction between several players. To the best of our knowledge, the distributed compression perspective developed in this paper has not been considered before.

1.6 Conclusions

We believe that the model raised is an extremely interesting one and is quite pertinent to the analyses of collective distributed phenomena where distributed entities are trying to come together to form joint actions. We believe the process and notation permit a much richer study, especially when one starts to allow correlations between the messages generated and the sender-receiver pairs. The ability to study the encoding and decoding functions – are they really functions, are they inverses of each other, how do they evolve? – are all intriguing questions that can now be subject to analyses. While our results do not address all these aspects, we do hope it will be the subject of future work.

In terms of the constructions and results, one interesting aspect of the compression protocol we use is that it mimics some of the curious features shown in human language. For every message m, player i and round t, the encoding function describes a specific word which is player i's encoding of m, i.e., it gives a (encoding) dictionary. The same player also possesses at the same round a decoding dictionary which we may view as saying, for every message m, which words this player would decode to m. Unlike in the basic schemes described, in our scheme the encoding dictionary is not identical to the decoding dictionary. While the encoding dictionary is a function mapping messages to words, the decoding dictionary is not: It is more conservative and lists many words for any given message. This phenomenon is definitely visible in human languages and our work suggests a plausible reason for the occurence of this phenomenon.

We now mention some important questions that arise from this work. On the conceptual side, it would be very interesting to further use the formalism and ideas developed in theoretical computer science over the last decades in order to capture the phenomena

exhibited by human languages. In particular, it would be interesting to extend our model to take into account other objectives along with compression. It would also be very interesting to consider the case where Q and the set of interacting players vary (slightly) with time, in the hope of modelling cultural changes that take place from one generation to another.

On the more technical side, we sticked in this work to the complete graph representing the interactions between various players. It would be worthwile to investigate other graph structures that favor the creation of communities, and study the properties of the language(s) that evolve in this case. Moreover, while we have considered in this work generic distributions Q, it would be nice to explore the data-structural aspects in the case where Q comes from a well-structured family of distributions (e.g, a Markov Chain). Finally, a concrete question is to determine whether the $O(\log(D(P||Q)))$ additive term in Theorem 3 is actually needed, which seems to be related to some intriguing questions about the chromatic number of certain families of graphs (see [4]).

Outline of the Rest of the Paper

In Section 2, we formally define our distributed compression model. In Section 3, we describe our main protocol along with its computationally efficient implementation (Section 3.1, Section 3.2, Section 3.3, Section 3.4 and Section 3.5). In Section 3.6, we describe a computationally inefficient variant of our protocol that requires smaller communication.

2 Formal Definitions

Throughout this paper, we denote by $H(Q) \triangleq \sum_x Q(x)\log(1/Q(x))$ the Shannon entropy of a probability distribution Q, and by $D(Q||P) \triangleq \sum_x Q(x)\log(Q(x)/P(x))$ the KL divergence between probability distributions Q and P. For any set S of elements, we write $u \in_R S$ to mean that u is sampled uniformly at random from the set S. We also denote by $\mathbb{N}$ the set of all natural numbers.

We now formally define our setup.

▶ **Definition 2** (Distributed Compression)**.** A *distributed compression protocol* Π is parametrized by a tuple (K, P, ε) where

- K is the number of players.
- P is a prior distribution over $\mathbb{N}$, which the players all agree on.
- ε is an error parameter.

The protocol is run on an instance parametrized by a pair (Q, T) where Q is the "true" distribution over $\mathbb{N}$, and T is the total number of iterations for which the protocol is run. Both Q and T are unknown to the players. In any iteration $t \in [T]$,

- Two distinct players i and j are chosen uniformly at random from $[K]$.
- A message m is sampled from distribution Q, and is given to player i.
- Player i attempts to communicate m to player j by sending a single message comprising of C_t bits.
- Player j outputs a message $\widehat{m}$.

The protocol is required to be such that, for any Q, and in any iteration t, it holds that $\Pr[\widehat{m} \neq m] \leq \varepsilon$, where the the probability is over the randomness of the messages and players chosen in the history of the protocol. The amortized communication cost of Π is defined to be $\sum_{t\in[T]} C_t/T$.

During the description of the protocol and the analysis, we will use t to denote the current iteration. Also, we will use t_i to denote the 'local time' of player i. That is, t_i is the number of times player i was picked as the sender. Note that $t = \sum_{i \in [K]} t_i$.

3 A Distributed Compression Protocol

In this section, we prove the following theorem, which is the same as Theorem 1 but "without the computational efficiency" part. The proof of the computational efficiency part of Theorem 1 appears in Section 3.5.

▶ **Theorem 3.** *Let $\varepsilon > 0$ be a sufficiently small positive absolute constant. For all K and P, there exists a deterministic distributed compression protocol Π, such that for any distribution Q over $\mathbb{N}$ when run for T iterations, the amortized communication cost of Π over T iterations approaches $O(H(Q) + \log D(Q||P) + \log(1/\epsilon))$ as T gets large.*

More formally, for $T \geq 8 \cdot 2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon) \cdot K$, the amortized communication cost is

$$O\left(H(Q) + \log D(Q||P) + \log(1/\epsilon) + \frac{2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon) \cdot K}{T} \cdot D(Q||P) + 1\right).$$

In the rest of this section, we describe the protocol behind the proof of Theorem 3.

3.1 Overview of the Protocol

We begin by giving a brief overview of the protocol. In any iteration, the chosen players will use one of two protocols that we call Static protocol (Section 3.2) and Uncertain protocol (Section 3.3).

On a high level, the Static protocol communicates messages with zero error, but it uses $H(Q) + D(Q||P) + O(1)$ bits of communication in expectation. On the other hand, the Uncertain protocol communicates $O(H(Q) + \log(D(Q||P)))$ bits in expectation, but it makes errors with some probability.

Suppose during iteration t, a message m is chosen to be sent by player i to player j, where m is sampled from the unknown distribution Q. In this case, player i will decide to communicate using either the Static protocol or the Uncertain protocol. Intuitively, in the initial few rounds in which player i is the sender, she will use the Static protocol as she does not want to risk incurring large error by using the Uncertain protocol. But, once player i has seen enough messages, she will switch to using the Uncertain protocol. The final bound on the amortized communication cost comes about by showing that the protocol ends up using the Uncertain protocol much more often than the Static protocol.

In Section 3.4, we describe exactly how the players switch between the two protocols and prove Theorem 3.

3.2 The Static Protocol

In the Static protocol, player i uses the Huffman codebook for distribution P in order to communicate the message m. The expected communication cost of doing so is $H(Q) + D(Q||P) + O(1)$. The good aspects of this protocol are that the error probability is zero, and the players do not require any knowledge about the unknown distribution Q. However, the downside is that the communication cost is quite high in terms of the dependence on $D(Q||P)$.

We summarize the Static protocol in the following straightforward lemma.

▶ **Lemma 4** (STATIC Protocol). *Suppose that during iteration t, a message m is chosen to be sent by player i to player j, where m is sampled according to the unknown distribution Q. Then, player i can communicate m to player j with zero error, such that the expected communication length is upper bounded by*

$$H(Q) + D(Q||P) + 1.$$

3.3 The Uncertain Protocol

The UNCERTAIN protocol is suitable when the players have individually learnt good estimates of the distribution Q. However, since the players do not exactly agree on their learned estimates, we need an approach for the players to communicate when their estimates of Q are *close* but may not be exactly identical. Our approach is inspired from [4], and we obtain a protocol that in expectation communicates roughly $O(H(Q) + \log D(Q||P)) + O(1)$ bits. We summarize the UNCERTAIN protocol in the following lemma,

▶ **Lemma 5** (UNCERTAIN Protocol). *Suppose that during iteration t, a message m is chosen to be sent by player i to player j, where m is sampled according to the unknown distribution Q. Then, player i can communicate m to player j, such that the expected communication length is upper bounded by*

$$O\left(H(Q) + \log D(Q||P) + \log(1/\epsilon) + 1\right).$$

Moreover, the error probability is at most

$$2 \cdot e^{-\frac{1}{8}\frac{Q(m)}{K}t} + \frac{\epsilon}{4},$$

where the randomness is over all past messages and players chosen in the previous iterations.

Isolating Hash Families

In order to describe the UNCERTAIN protocol achieving Lemma 5, we will need the following notion of an *isolating hash family* which generalizes that of [4].

▶ **Definition 6** (Isolating Hash Families). Let N, R and ℓ be positive integers and $\varepsilon \in (0, 1]$. Then, a collection $\mathcal{H} = \{h_1, h_2, \ldots, h_M : [N] \to [R]\}$ is said to be (N, ℓ, ϵ)-isolating if for every subset $S \subseteq [N]$ with $|S| \leq 2^{\ell-1}$ and every $m \in [N] \setminus S$, we have that $\Pr_{h\in\mathcal{H}}[h(m) \in h(S)] < \varepsilon$. We call M the *size* and R the *range-size* of the isolating hash family $\mathcal{H}$. The family $\mathcal{H}$ is said to be *efficiently computable* if there is an algorithm that takes as input $i \in [M]$ and $j \in [N]$ and computes $h_i(j)$ in time polynomial in $\log M$, $\log N$ and $\log R$.

We note that the family used in [4] corresponds to setting $\epsilon = 1$ in Definition 6. The next lemma shows the existence of an *explicit* and *efficiently computable* (N, ℓ, ϵ)-isolating hash family of relatively small size and small range-size.

▶ **Lemma 7.** *For every positive integers N and ℓ and every $\varepsilon \in (0, 1]$, there exists an* explicit *and* efficiently computable *(N, ℓ, ε)-isolating hash family $\mathcal{H}_{(N,\ell,\varepsilon)}$ of size and range-size at most $2^\ell \cdot \frac{\log N}{\varepsilon}$.*

Proof. Let $q = 2^{\ell+\lceil \log n + \log \frac{1}{\varepsilon}\rceil}$. For each $x \in \mathbb{F}_q$, define the function h_x to be the evaluation of the polynomial defined by m on x, i.e.,

$$h_x(m_0, ..., m_{n-1}) \triangleq \sum_{i=0}^{n-1} m_i x^i.$$

By the fundamental theorem of algebra, for every $m' \neq m$, we have that $\Pr_x[h_x(m) = h_x(m')] \leq \frac{n}{q} \leq 2^{-\ell - \log \frac{1}{\varepsilon}}$. Thus, by the union bound, for every set S of size at most $2^{\ell-1}$, we have $\Pr[f(m) \in f(S)] \leq \varepsilon$, as required. ◀

Pre-Processing Step

As stated earlier, all the players come in with a prior distribution P. In addition, as part of the pre-processing, they compute and store the following:

- Divide the input space $\mathbb{N}$ into a countable number of buckets indexed by $r \in \mathbb{N}_{>0}$, given by $A_r = \left\{m : 2^{-r} < P(m) \leq 2^{-r+1}\right\}$. Clearly, for any r, it holds that $|A_r| \leq 2^r$. In addition, define the function $r(m) := \lceil \log(1/P(m)) \rceil$ for every $m \in \mathbb{N}$, that is, $r(m)$ is the index of the bucket to which m belongs.
- For every r, fix an (arbitrary) choice of isolating hash families $\mathcal{H}_{(N,\ell,\epsilon/4)}$, for $N = |A_r|$ and every choice of $\ell \in \{1, 2, \cdots, \lceil \log N \rceil\}$.

Suppose during iteration t, a message m is chosen to be sent by player i to player j, where m is sampled according to the unknown distribution Q. Define Q_t^i to be the empirical distribution of the samples seen by player i up to iteration t (which includes the iteration t, where the message seen is m). Similarly, define Q_t^j to be the empirical distribution of the samples seen by player j up to iteration t (this includes iteration t, but by definition player j does not see any message in this iteration). The players use the encoding and decoding strategies described next.

Encoding

Upon receiving message m, player i does the following,

(i) let $A \stackrel{\text{def}}{=} A_{r(m)}$ and $N \stackrel{\text{def}}{=} |A|$.
(ii) let $\ell = \left\lceil \log(4/Q_t^i(m)) \right\rceil$.
(iii) let $u \in_R [\mathcal{H}_{(N,\ell,\epsilon/4)}]$.
(iv) Send the tuple $(r, \ell, u, h_u(m))$ to player j.

The intuition for this encoding is as follows: upon receiving r, player j understands that $m \in A_r$, upon receiving ℓ, she understands which hash family to use, upon receiving u, she knows which hash function to use, and hopefully with $h_u(m)$, she will be able to recover m correctly.

Decoding

Upon receiving the tuple (r, ℓ, u, h_*), player j does the following:

(i) Set $A = A_r$ and $N \stackrel{\text{def}}{=} |A|$.
(ii) Identify $h_u \in \mathcal{H}_{(N,\ell,\epsilon/4)}$.
(iii) Output $\arg\max_{m' \in A : h_u(m') = h_*} Q_t^j(m')$.

3.3.1 Analysis

We now analyze the operation of the above protocol.

Communication Cost

Suppose the message m is chosen to be sent by player i to player j. The communication cost of sending the tuple $(r, \ell, u, h_u(m))$ is as follows:

(i) $\log \lceil \log(1/P(m)) \rceil$ bits to send r.

(ii) $\log(\log(1/Q_t^i(m)) + 3)$ bits to send ℓ, since $\ell \le \log|S| + 1 \le \log(4/Q_t^i(m)) + 1$.
(iii) $\log(1/Q_t^i(m)) + \log\lceil\log(1/P(m))\rceil + \log(1/\epsilon) + 5$ bits to send u (it takes $\ell + \log\log N + \log(4/\epsilon)$ bits).
(iv) $\log(1/Q_t^i(m)) + \log\lceil\log(1/P(m))\rceil + \log(1/\epsilon) + 5$ bits to send $h_u(m)$.
Thus, the total communication is given by,

$$2\underbrace{\log(1/Q_t^i(m))}_{(I)} + 3\underbrace{\log\lceil\log(1/P(m))\rceil}_{(II)} + \underbrace{\log(\log(1/Q_t^i(m)) + 3) + 10}_{(III)} + 2\log(1/\epsilon)$$

We wish to prove guarantees on the expected communication cost, when m is drawn from Q. The terms in (III) are lesser order terms, which are smaller than (I), thus we can ignore them. Term (II) in expectation is,

$$\mathop{\mathbb{E}}_{m\sim Q}\left[\log\left(\left\lceil\log\frac{1}{P(m)}\right\rceil\right)\right] \le \log\left(\mathop{\mathbb{E}}_{m\sim Q}\left[\log\frac{1}{P(m)}\right]\right) \le \log(H(Q) + D(Q||P) + 1)$$

Term (I) is slightly more tricky to bound in expectation. Note that the empirical distribution changes on receiving message m (this turns out to be critical in bounding the communication!). That is, $Q_t^i(m) = \frac{1+(t-1)Q_{(t-1)}^i(m)}{t}$. Also let $\mathcal{M}_t^i$ be the multi-set of all messages that player i has seen up to time t. Thus, Term (I) in expectation is as follows,

$$\mathop{\mathbb{E}}_{\mathcal{M}_{(t-1)}^i}\mathop{\mathbb{E}}_{m\sim Q}\left[\log\frac{1}{Q_t^i(m)}\right] = H(Q) + \mathop{\mathbb{E}}_{\mathcal{M}_{(t-1)}^i}\mathop{\mathbb{E}}_{m\sim Q}\left[\log\frac{Q(m)}{\frac{1}{t_i} + \frac{(t_i-1)Q_{(t-1)}^i(m)}{t_i}}\right]$$

In order to bound the second term above, we consider two cases, (i) $Q_{(t-1)}^i(m) \ge Q(m)/2$ or (ii) $Q_{(t-1)}^i(m) < Q(m)/2$. After fixing t_i and m, by Chernoff bound over the randomness of $\mathcal{M}_{(t-1)}^i$ we have that case (i) happens with probability at least $1 - \exp(-t\cdot Q(m)/8)$.

$$\text{Case (i) } Q_{(t-1)}^i(m) \ge Q(m)/2 \implies \log\left(\frac{Q(m)}{\frac{1}{t_i} + \frac{(t-1)Q_{(t)}^i(m)}{t_i}}\right) \le 1$$

$$\text{Case (ii) } Q_{(t-1)}^i(m) < Q(m)/2 \implies \log\left(\frac{Q(m)}{\frac{1}{t_i} + \frac{(t_i-1)Q_{(t-1)}^i(m)}{t_i}}\right) \le \log(t_i\cdot Q(m))$$

Using these upper bounds we get that,

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathcal{M}_{(t-1)}^i}\mathop{\mathbb{E}}_{m\sim Q}\left[\log\frac{Q(m)}{\frac{1}{t} + \frac{(t_i-1)Q_{(t-1)}^i(m)}{t_i}}\right] &\le \mathop{\mathbb{E}}_{m\sim Q}\left[1\cdot\left(1 - e^{-t_i\cdot Q(m)/8}\right) + \log(t_i\cdot Q(m))\cdot e^{-t_i\cdot Q(m)/8}\right] \\ &\le 1 + \mathop{\mathbb{E}}_{m\sim Q}\left[\log(t_i\cdot Q(m))\cdot e^{-t_i\cdot Q(m)/8}\right] \\ &\le 2\,,\end{aligned}$$

where the last inequality just follows from the fact that $\log(x)\cdot e^{-x/8} \le 1$ for all x.
Thus the overall communication is bounded by

$$(2 + o(1))\,H(Q) + 3\log D(Q||P) + 2\log(1/\epsilon) + O(1)\,.$$

Error Guarantee

We now show that the error probability in iteration t, denoted by p_t^{err} of the protocol is upper bounded by $2\cdot e^{-\frac{1}{8}\frac{Q(m)}{K}t} + \epsilon/4$, where m is fixed to be the message sent in round t.

Since player i has communicated (r, ℓ, u), player j knows the correct bucket of messages A_r to which m belongs. Knowing ℓ and u, player j also knows which hash function is being used, which is chosen to ensure that for every set S of size $\leq 2^\ell$, with probability $1 - \varepsilon/4$, for all $m' \in S \setminus \{m\}$, $h_u(m) \neq h_u(m')$.

Thus, if $\ell \leq \log(1/Q_t^j(m))$ then the j-th player will distinguish m from the set $S = \{m' \in A \mid Q_t^j(m') \geq Q_t^j(m)\}$ with probability $1 - \varepsilon/4$. We will bound the probability that this does not happen.

$$\begin{aligned} p_t^{\text{err}} &\leq \Pr\left[\ell > \log(1/Q_t^j(m))\right] + \frac{\epsilon}{4} \\ &\leq \Pr\left[4 \cdot Q_t^i(m) \geq Q_t^j(m)\right] + \frac{\epsilon}{4} \\ &\leq \Pr\left[Q_t^i(m) \geq 2 \cdot Q(m)\right] + \Pr\left[Q_t^j(m) \leq \frac{1}{2}Q(m)\right] + \frac{\epsilon}{4} \\ &\leq e^{-\frac{1}{3}\frac{Q(m)}{K}t} + e^{-\frac{1}{8}\frac{Q(m)}{K}t} + \frac{\epsilon}{4}\,, \end{aligned}$$

where the last equality follows by Chernoff bound and the fact that Q_t^i, Q_t^j are binomial distributions with parameters t and $\frac{Q(m)}{K}$.

3.4 Final Protocol

We are now ready to present the protocol desired in Theorem 3. As before, suppose that during iteration t, a message m is chosen to be sent by player i to player j, where m is sampled according to the unknown distribution Q. As defined in Section 3.3, define Q_t^i to be the empirical distribution of the samples seen by player i up to iteration t (which includes the iteration t, where the message seen is m). Similarly define Q_t^j to be the empirical distribution of the samples seen by player j up to iteration t (this includes iteration t, but by definition player j does not see any message in this iteration).

For ease of presentation, we will first assume that the players know the entropy of the distribution Q. This is not a natural assumption, and indeed we do get around it in Section 3.4.2. However, we will describe the main protocol with this assumption to make the analysis more intuitive.

Encoding

Upon receiving message m, player i does the following:

- If $t_i < 80 \cdot 2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon)$,
  - send the bit $b = 0$
  - use the Static protocol (Lemma 4) to send message m.
- Else,
  - send the bit $b = 1$
  - use the Uncertain protocol (Lemma 5) to send message m.

(where the bit b indicates whether player i is using the Static protocol or the Uncertain protocol).

Decoding

Depending on the value of the received bit b, player j uses either the Static protocol or the Uncertain protocol to decode and output $\widehat{m}$.

3.4.1 Analysis

We now upper-bound the amortized communication cost and the error probability in any iteration of the above protocol.

Communication Cost

By the design of the final protocol, each player uses the STATIC protocol at most $80 \cdot 2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon)$ times, and hence overall, the STATIC protocol is used at most $O(2^{2H(Q)/\varepsilon} \cdot \log(1/\varepsilon) \cdot K)$ times. Thus, if the total number of iterations is T, then the total communication cost in expectation is at most,

$$\underbrace{O\left(2^{2H(Q)/\varepsilon} \cdot \log(1/\varepsilon) \cdot K\right) \cdot (H(Q) + D(Q||P) + 1)}_{\text{STATIC}} + \underbrace{T \cdot O\Big(H(Q) + \log D(Q||P) + O(1)\Big)}_{\text{UNCERTAIN}} .$$

And hence, the expected amortized communication cost is at most

$$O\left(H(Q) + \log D(Q||P) + \frac{2^{2H(Q)/\varepsilon} \cdot \log(1/\varepsilon) \cdot K}{T} \cdot D(Q||P) + 1\right).$$

Error Guarantee

We first show the following lemma, which is an easy consequence of Markov's inequality.

▶ **Lemma 8.** *For any distribution Q over $\mathbb{N}$, it holds that,*

$$\Pr_{m \sim Q}\left[Q(m) \geq 2^{-H(Q)/\varepsilon}\right] \geq 1 - \varepsilon.$$

Proof. By the definition of the entropy $H(Q)$, we have that $\mathbb{E}_{m \sim Q}\left[\log \frac{1}{Q(m)}\right] = H(Q)$. Thus, the following application of Markov's inequality immediately implies the lemma:

$$\Pr_{m \sim Q}\left[\log \frac{1}{Q(m)} \geq \frac{H(Q)}{\varepsilon}\right] \leq \varepsilon.$$ ◀

We will show that in any iteration t, the error probability is at most ε, where the randomness is over all the past and current messages and chosen players. We distinguish two cases:

Case 1. If $t < 8 \cdot 2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon) \cdot K$:
Using the Chernoff bound, it is easy to see that

$$\Pr\left[t_i > 80 \cdot 2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon) \;\middle|\; t < 2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon) \cdot K\right] \leq \exp\left[-\Omega\left(2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon)\right)\right] \ll \varepsilon.$$

Thus, it follows that with probability $\geq 1 - \varepsilon$, player i uses the STATIC protocol in which case there is zero error. Thus, the probability of error is at most ε.

Case 2. If $t \geq 8 \cdot 2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon) \cdot K$:
Section 8 implies that when a message m is sampled from Q, with probability at least $1 - \varepsilon/2$ it holds that $Q(m) \geq 2^{-2H(Q)/\varepsilon}$. In this situation, player i may choose to use either the STATIC or the UNCERTAIN protocol. In the former case, the protocol makes no error. In the latter case, by Lemma 5, the protocol makes error with probability at most

$$2 \cdot e^{-\frac{1}{8} K \frac{t}{Q(m)}} + \frac{\epsilon}{4} \leq 2 \cdot e^{-\frac{1}{8} 2^{-2H(Q)/\varepsilon} 2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon)} + \frac{\epsilon}{4},$$

which is at most $\varepsilon/2$ if $Q(m) \geq 2^{-2H(Q)/\varepsilon}$. Hence, the total error probability is at most ε.

3.4.2 Getting around the entropy assumption

We let $\varepsilon > 0$ be a sufficiently small positive absolute constant. We now informally describe how to construct a protocol that does not assume that the players know the entropy of the distribution Q. We note that the main reason for the "switching" criterion $t_i < 80 \cdot 2^{2H(Q)/\varepsilon} \cdot \log(8/\varepsilon)$ was to ensure that when we are using the UNCERTAIN protocol and we encounter a message m with $Q(m) \geq 2^{-2H(Q)/\varepsilon}$ (which happens with probability at least $1 - \varepsilon/2$), it holds that $t_i \cdot Q(m) \gg \log(1/\varepsilon)$.

Thus, the protocol guarantees will still hold as long as the players switch to the UNCERTAIN protocol after a sufficiently "large" time t_i. Indeed, we show that it is possible to switch to the UNCERTAIN protocol after time t_i such that $\Pr_{m\sim Q}\left[t_i \cdot Q(m) \gg \log(1/\varepsilon)\right] \geq 1 - \frac{\varepsilon}{4}$.

We now describe the "switching" criterion. In what follows, we prove that for every player, the switching criterion is not met too early, nor is it met too late. Lemma 10 shows that the probability that the switching criterion is met "too early" (i.e., before the time T_0 defined below) is very small. Moreover, it turns out that the probability that the switching criterion is met "too late" (i.e., after time $2^{O\left(\frac{H(Q)}{\epsilon}\right)} \cdot K$) is also very small (see Lemma 9 below). Together, these two properties allow individual players to switch from the STATIC protocol to the UNCERTAIN protocol based on their observed history of messages. In turn, this allows us to carry out an analysis of the communication cost and the error probability without knowledge of the entropy of Q.

We say that at player i, the switching criterion is met at iteration t_i if

$$t_i \geq \varepsilon^{-3} \quad \text{and} \quad \sum_{m:Q_t^i(m) > t_i^{-\frac{1}{2}}} Q_t^i(m) \geq 1 - \frac{\epsilon}{2}.$$

We first show that, with high probability, the switching criterion is met in time $2^{O\left(\frac{H(Q)}{\epsilon}\right)} \cdot K$

▶ **Lemma 9.** *For every player i, the probability that the switching criterion is met before time $t > 4 \cdot 2^{\frac{16H(Q)}{\epsilon}} K$ is at least $1 - \exp\left(-\frac{1}{64}\epsilon^2 2^{-\frac{4H(Q)}{\epsilon}} \frac{t}{K}\right)$.*

Proof. Let m be such that $Q(m) \geq 2^{-\frac{4H(Q)}{\epsilon}}$. By the Chernoff bound,

$$\Pr\left[Q_t^i(m) \leq (1 - \frac{\epsilon}{4})Q(m)\right] \leq \exp\left(-\frac{1}{32}\epsilon^2 \frac{Q(m)}{K} t\right).$$

Moreover, by the Chernoff bound, we have that $\Pr[t_i \leq \frac{t}{2K}] \leq \exp\left(-\frac{t}{8K}\right)$. We define the event

$$E = \left[t_i \leq \frac{t}{2K} \vee \exists m : Q(m) \geq 2^{-\frac{4H(Q)}{\epsilon}} \wedge Q_t^i(m) \leq (1 - \frac{\epsilon}{4})Q(m)\right].$$

By the union bound, we get that

$$\begin{aligned}
\Pr[E] &\leq \exp\left(-\frac{t}{8K}\right) + \exp\left(\frac{4H(Q)}{\epsilon} - \frac{1}{32}\epsilon^2 \frac{2^{-\frac{4H(Q)}{\epsilon}}}{K} t\right) \\
&\leq \exp\left(-\frac{1}{64}\epsilon^2 2^{-\frac{4H(Q)}{\epsilon}} \frac{t}{K}\right).
\end{aligned}$$

If the event E does not hold, then for every m that satisfies $Q(m) > 2^{-\frac{4H(Q)}{\epsilon}}$ we get that

$$Q_t^i(m) > (1 - \frac{\epsilon}{4})Q(m) > (1 - \frac{\epsilon}{4})2^{-\frac{4H(Q)}{\epsilon}} > \left(\frac{t}{2K}\right)^{-\frac{1}{2}} > t_i^{-\frac{1}{2}}.$$

Thus,

$$\begin{aligned}
\sum_{m:Q_t^i(m)>t_i^{-\frac{1}{2}}} Q_t^i(m) &\geq \sum_{m:Q(m)>2^{-\frac{4H(Q)}{\epsilon}}} Q_t^i(m) \\
&\geq (1-\frac{\epsilon}{4})\cdot \sum_{m:Q(m)>2^{-\frac{4H(Q)}{\epsilon}}} Q(m) \\
&= (1-\frac{\epsilon}{4})\cdot \Pr_{m\sim Q}\left[Q(m)>2^{-\frac{4H(Q)}{\epsilon}}\right] \\
&= (1-\frac{\epsilon}{4})\cdot \Pr_{m\sim Q}\left[\log\frac{1}{Q(m)}<\frac{4H(Q)}{\epsilon}\right] \\
&\geq (1-\frac{\epsilon}{4})\cdot(1-\frac{\epsilon}{4}) \\
&\geq 1-\frac{\epsilon}{2}.
\end{aligned}$$

Moreover, $t_i > \frac{t}{2K} > \varepsilon^{-3}$. Hence, in this case, the switching criterion is met. ◀

Let T_0 be the smallest $t > \frac{1}{\varepsilon^3 K}$ that satisfies $\Pr_{m\sim Q}\left[Q(m) \geq \frac{1}{4}\sqrt{\frac{K}{t}}\right] > 1-\frac{3}{4}\epsilon$. First, we will observe that after time T_0, it is indeed safe to switch to the UNCERTAIN protocol.

▶ **Observation 3.1.** *For every time $t \geq T_0$, the* UNCERTAIN *protocol succeeds with probability at least $1-\varepsilon$.*

Proof. By Lemma 5, with probability at least $1-\frac{3}{4}\varepsilon$, the protocol succeeds with probability at least $1-\frac{\varepsilon}{4}$. ◀

It remains to show that with high probability, we will not use the Uncertain protocol before T_0.

▶ **Lemma 10.** *The probability that player i meet the switching criterion before time T_0 is at most ε.*

Proof. We will show that for any fixed $t_i \leq \frac{2T_0}{K}$, we have that the probability that for player i, the switching criterion is met in local time t_i, is at most $2\cdot\exp\left(-\frac{1}{12}\sqrt{t_i}\right)$. By the union bound, we will get that the probability that for player i, the switching criterion is met before local time $\frac{2T_0}{K}$ is bounded by

$$\sum_{t_i=\epsilon^{-3}+1}^{\infty} 2\cdot\exp\left(-\frac{1}{12}\sqrt{t_i}\right) \leq \int_{\epsilon^{-3}}^{\infty} 2\cdot\exp\left(-\frac{1}{12}\sqrt{t_i}\right)dt_i = 24\cdot(\sqrt{\epsilon^{-3}}+12)\cdot\exp\left(-\frac{1}{12}\sqrt{\epsilon^{-3}}\right) \leq \frac{\epsilon}{2}.$$

Moreover, by the Chernoff bound, we have that the probability that the local time of player i in (global) time T_0 exceeds $\frac{2T_0}{K}$ is at most $\exp(-\frac{T_0}{3K})$. Thus, the probability that for player i the switching criterion is met before time T_0 is at most $\frac{\varepsilon}{2}+\exp(-\frac{T_0}{3K}) \leq \epsilon$, as required.

Fix t_i and let $M = \left\{m\in\mathbb{N} \mid Q(m) \geq \frac{1}{2\sqrt{t_i}}\right\}$. Since $t_i < \frac{2T_0}{K}$, we have that

$$\Pr_{m\sim Q}[m\in M] = \Pr_{m\sim Q}\left[Q(m)\geq\frac{1}{2\sqrt{t_i}}\right] \leq \Pr_{m\sim Q}\left[Q(m)\geq\frac{\sqrt{K}}{2\sqrt{2T_0}}\right] \leq \Pr_{m\sim Q}\left[Q(m)\geq\frac{\sqrt{K}}{4\sqrt{T_0}}\right] \leq 1-\frac{3}{4}\epsilon\,.$$

Thus, by the Chernoff bound,

$$\Pr\left[\sum_{m\in M} Q_t^i(m) \geq 1-\frac{\varepsilon}{2}\right] \leq \exp\left(-\frac{\epsilon\cdot t_i}{25}\right) \tag{1}$$

Now we upper bound $\Pr\left[\exists m \notin M : Q_t^i(m) > \frac{1}{2}\sqrt{\frac{K}{t}}\right]$. To prove this bound, we can assume without lost of generality that for all m except one, we have that $Q(m) > \frac{1}{5\sqrt{t_i}}$: if there exist two elements of such a small probability, we can merge them together to a single element and only increase the probability $\Pr\left[\exists m \notin M : Q_t^i(m) > \frac{1}{2}\sqrt{\frac{K}{t}}\right]$. So we will assume that there are at most $5\sqrt{t_i}+1$ such elements. By the Chernoff bound, we have that for each $m \notin M$, $\Pr\left[Q_t^i(m) > \frac{1}{\sqrt{t_i}}\right] \leq \exp\left(-\frac{1}{6}\sqrt{t_i}\right)$ and by a union bound we can get that

$$\Pr\left[\exists m \notin M : Q_t^i(m) > \frac{1}{\sqrt{t_i}}\right] \leq (5\sqrt{t_i}+1)\cdot\exp\left(-\frac{1}{6}\sqrt{t_i}\right) \leq \exp\left(-\frac{1}{12}\sqrt{t_i}\right). \tag{2}$$

By Combining Equations 1 and 2 , assuming $t_i \geq \varepsilon^{-3}$, we get

$$\Pr\left[\sum_{m:Q_t^i(m)>\frac{1}{\sqrt{t_i}}} Q_t^i(m)\right] \leq \Pr\left[\sum_{m\in M} Q_t^i(m) \geq 1-\frac{\epsilon}{2} \vee \exists m \notin M : Q_t^i(m) > \frac{1}{\sqrt{t_i}}\right] \leq 2\cdot\exp\left(-\frac{1}{12}\sqrt{t_i}\right).$$

This gives an upper bound of $2\cdot\exp\left(-\frac{1}{12}\sqrt{t_i}\right)$ on the probability that at player i, the switching criterion is met in local time t_i, as needed. ◀

3.5 Efficient Implementation

We briefly sketch how to *efficiently* implement the encoding and decoding strategies of Section 3. The details are deferred to the full version. The overall update time will be linear in $(H(Q)+D(Q||P))/\epsilon$, and the used memory will be proportional to the dictionary-size which is exponential in $(H(Q)+D(Q||P))/\epsilon$. The key question of interest is how to compute the uncertain compression function efficiently. Note that while we would like a fast "processing time" per update, the model naturally allows us to amortize the cost over many operations. In particular, the switch from the STATIC protocol to the UNCERTAIN one does not have to be carried out in an instant. We will exploit this feature strongly. The corresponding efficient algorithm will have three phases:

1. A phase where we simply use the STATIC protocol while updating the empirical distributions.
2. A phase where the encoding and decoding dictionaries are being built, but where we still use the STATIC protocol.
3. A phase where we use the UNCERTAIN protocol.

In what follows, we assume that the messages m and the prior distribution P are presented jointly so that the message m given to player i in round t is $E_P(m)$, namely the STATIC (Huffman) encoding of m under P. This is a natural assumption about P – after all P is meant to represent a simple and natural, though unoptimized, distribution over the message space. We now recall the statement of Theorem 1.

▶ **Theorem 1.** *Let $\varepsilon > 0$ be a sufficiently small positive absolute constant. For all K and P, there exists a deterministic distributed compression protocol Π, such that for any distribution Q over $\mathbb{N}$ when run for T iterations,*

- *the amortized communication cost of Π over T iterations approaches $O(H(Q)+\log D(Q||P)+\log(1/\epsilon))$ as T gets large. More formally, the amortized communication cost is*

$$O\left(H(Q)+\log D(Q||P)+\log(1/\epsilon)+\frac{2^{\Theta(H(Q)+D(Q||P))/\varepsilon}\cdot K}{T}\cdot D(Q||P)+1\right)$$

- *in each round, the transmitter and receiver run in time linear in their input and output sizes.*
- *the space usage is exponential in* $(H(Q) + D(Q||P))/\epsilon$.

Note that in Theorem 1, the input to the transmitter is $E_P(m)$ and the input to the receiver is the message that she gets from the transmitter.

Proof Sketch. Let $T_\epsilon = 2^{\Theta(H(Q)+D(Q||P))/\epsilon}$ denote the local time at which our inefficient transmitter and receiver – described in the previous section – should switch from the STATIC protocol to the UNCERTAIN one. In the efficient protocol, during the execution of the STATIC protocol for the first T_ϵ units of local time, each player will also maintain a count of the number of times she has seen each message using a simple binary tree indexed by $E_P(m)$. At local time T_ϵ, player i updates his empirical distribution $Q^i_{T_\epsilon}$. Note that we can amortize this update time over several rounds. After round T_ϵ, the efficient protocol will start building an encoding and decoding table for the uncertain compression algorithm, but will take $T' = \mathsf{poly}(T_\epsilon)$ rounds to do so (as we will explain below), and in the meanwhile, it will continue using the STATIC protocol for these T' rounds. At round $T_\epsilon + T'$, it will then switch to the UNCERTAIN protocol, and at this stage it will have a complete table (for all relevant messages) for the encoding and decoding functions, and so it can encode and decode by a simple table lookup.

We also note that the upper bound on the amortized communication cost follows from a similar argument as in the proof of Theorem 3 in Section 3.

So it suffices to show that the encoding and decoding tables can be computed in time $\mathsf{poly}(T_\epsilon)$. A straightforward implementation of the algorithm used in the proof of Theorem 3 essentially works, with a few additional observations. First, we note that we do not need to encode messages m with $P(m) \leq 2^{-\Theta(H(Q)+D(Q||P))/\epsilon}$ since by Markov's inequality such messages occur with probability less than ϵ. This makes sure that the hash families that we need work with a value of N which is at most $2^{(H(Q)+D(Q||P))/\epsilon}$ and the $\log N$ factor in the size of these hash families is equal to $(H(Q) + D(Q||P))/\epsilon$, which is affordable. Next, we use the efficiently computable hash functions which are given by Lemma 7. We apply these hash functions to $E_P(m)$ rather than m in order to make sure that their domain is also small. The upper bound on the encoding time now follows.

For the decoding time, we note that filling in one entry of the decoding table takes time linear in N which is exponentially larger than the budget in the statement of Theorem 1. However, we can divide this task over N rounds while performing $O(1)$ computations per round. The upper bound on the decoding time now follows.

Finally, the space usage is proportional to the size of the encoding and decoding lookup tables which is exponential in $(H(Q) + D(Q||P))/\epsilon$. ◀

3.6 A Computationally Inefficient Protocol with Smaller Communication

In this section, we show that if one does not care about computational efficiency, then we can remove the additive $\log(1/\epsilon)$ term from the communication cost in Theorem 1 while also replacing the multiplicative $2^{\Theta(H(Q)+D(Q||P))/\varepsilon}$ factor by $2^{\Theta(H(Q))/\varepsilon}$. The details are deferred to the full version.

The general structure of the protocol is similar to the one in Section 3 except that for the description and analysis of the UNCERTAIN protocol (Section 3.3). We now describe a computationally inefficient variant of the UNCERTAIN protocol which has smaller communication. The performance of this variant is summarized in the following lemma.

▶ **Lemma 11** (Uncertain Protocol). *Suppose that during iteration t, a message m is chosen to be sent by player i to player j, where m is sampled according to the unknown distribution Q. Then, player i can communicate m to player j, such that the expected communication length is upper bounded by*

$$O\left(H(Q) + \log D(Q||P)\right) + O(1).$$

Morever, the error probability is at most

$$\frac{1}{Q(m)} \cdot \exp\left(-\Omega\left(\frac{t \cdot Q(m)}{K}\right)\right),$$

where the randomness is over all past messages and players chosen in the previous iterations.

We now describe the corresponding encoding and decoding procedures (along with the pre-processing step). Recall Definition 6 of an (N, ℓ, ϵ)-isolating hash family. We now define an (N, ℓ)-isolating hash family to be an $(N, \ell, 1)$-isolating hash family.

Pre-Processing Step

As stated earlier, all the players come in with a prior distribution P. In addition, as part of the pre-processing, they compute and store the following:

- Divide the input space $\mathbb{N}$ into a countable number of buckets indexed by $r \in \mathbb{N}_{>0}$, given by $A_r = \left\{m : 2^{-r} < P(m) \leq 2^{-r+1}\right\}$. Clearly, for any r, it holds that $|A_r| \leq 2^r$. In addition, define the function $r(m) := \lceil \log(1/P(m)) \rceil$ for every $m \in \mathbb{N}$, that is, $r(m)$ is the index of the bucket to which m belongs.
- For every r, fix an (arbitrary) choice of isolating hash families $\mathcal{H}_{(N,\ell)}$, for $N = |A_r|$ and every choice of $\ell \in \{1, 2, \cdots, \lceil \log N \rceil\}$.

Suppose during iteration t, a message m is chosen to be sent by player i to player j, where m is sampled according to the unknown distribution Q. Define Q_t^i to be the empirical distribution of the samples seen by player i up to iteration t (which includes the iteration t, where the message seen is m). Similarly, define Q_t^j to be the empirical distribution of the samples seen by player j up to iteration t (this includes iteration t, but by definition player j does not see any message in this iteration). The players use the encoding and decoding strategies described next.

Encoding

Upon receiving message m, player i does the following,

(i) let $A \stackrel{\text{def}}{=} A_{r(m)}$ and $N \stackrel{\text{def}}{=} |A|$.
(ii) let $S \stackrel{\text{def}}{=} \left\{m' \in A \setminus \{m\} : Q_t^i(m') \geq \frac{1}{16} Q_t^i(m)\right\}$.
(iii) let $\ell = \lceil \log |S| \rceil$.
(iv) let $u \in [|\mathcal{H}_{(N,\ell)}|]$ and $h_u \in \mathcal{H}_{(N,\ell)}$ such that $h_u(m) \notin h_u(S)$.
(v) Send the tuple $(r, \ell, u, h_u(m))$ to player j.

Note that the property of isolating hash families (see Definition 6) guarantees the existence of $h_u \in \mathcal{H}_{(N,\ell)}$ as desired in (iv).

The intuition for this encoding is as follows: upon receiving r, player j understands that $m \in A_r$, upon receiving ℓ, she understands which hash family to use, upon receiving u, she knows which hash function to use, and hopefully with $h_u(m)$, she will be able to recover m correctly.

Decoding

Upon receiving the tuple (r, ℓ, u, h_*), player j does the following:

(i) Set $A = A_r$ and $N \stackrel{\text{def}}{=} |A|$.
(ii) Identify $h_u \in \mathcal{H}_{(N,\ell)}$.
(iii) Output $\arg\max_{m' \in A : h_u(m') = h_*} Q_t^j(m')$.

The analysis of the communication cost and the error guarantee appears in Appendix A, where Lemma 11 is proved.

References

1 Douglas Adams. *The Hitchhiker's Guide to the Galaxy #1*. Del Rey, 1979.

2 Noam Chomsky. Reflections on language. *New York*, 3, 1975.

3 Noam Chomsky. Rules and representations. *Behavioral and brain sciences*, 3(01):1–15, 1980.

4 Elad Haramaty and Madhu Sudan. Deterministic compression with uncertain priors. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 377–386. ACM, 2014.

5 Marc D Hauser, Noam Chomsky, and W Tecumseh Fitch. The faculty of language: what is it, who has it, and how did it evolve? *science*, 298(5598):1569–1579, 2002.

6 Brendan Juba, Adam Tauman Kalai, Sanjeev Khanna, and Madhu Sudan. Compression without a common prior: an information-theoretic justification for ambiguity in language. In *Innovations in Computer Science - ICS*, pages 79–86. Tsinghua University Press, 2011.

7 Simon Kirby, Mike Dowman, and Thomas L Griffiths. Innateness and culture in the evolution of language. *Proceedings of the National Academy of Sciences*, 104(12):5241–5245, 2007.

8 Erez Lieberman, Jean-Baptiste Michel, Joe Jackson, Tina Tang, and Martin A Nowak. Quantifying the evolutionary dynamics of language. *Nature*, 449(7163):713–716, 2007.

9 Martin A Nowak. Evolutionary biology of language. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 355(1403):1615–1622, 2000.

10 Martin A Nowak and Natalia L Komarova. Towards an evolutionary theory of language. *Trends in cognitive sciences*, 5(7):288–295, 2001.

11 Martin A Nowak, Natalia L Komarova, and Partha Niyogi. Computational and evolutionary aspects of language. *Nature*, 417(6889):611–617, 2002.

12 Martin A Nowak and David C Krakauer. The evolution of language. *Proceedings of the National Academy of Sciences*, 96(14):8028–8033, 1999.

13 Martin A Nowak, Joshua B Plotkin, and David C Krakauer. The evolutionary language game. *Journal of Theoretical Biology*, 200(2):147–162, 1999.

14 Steven Pinker and Paul Bloom. Natural language and natural selection. *Behavioral and brain sciences*, 13(04):707–727, 1990.

15 Joshua B Plotkin and Martin A Nowak. Language evolution and information theory. *Journal of Theoretical Biology*, 205(1):147–159, 2000.

16 Giacomo Rizzolatti and Michael A Arbib. Language within our grasp. *Trends in neurosciences*, 21(5):188–194, 1998.

A Analysis of Computationally Inefficient Protocol

We now analyze the operation of the protocol described in Section 3.6.

Communication Cost

Suppose the message m is chosen to be sent by player i to player j. The communication cost of sending the tuple $(r, \ell, u, h_u(m))$ is as follows:

(i) $\log\lceil\log(1/P(m))\rceil$ bits to send r.
(ii) $\log(\log(1/Q^i_t(m)) + 5)$ bits to send ℓ, since $\ell \le \log|S| + 1 \le \log(16/Q^i_t(m)) + 1$.
(iii) $\log(1/Q^i_t(m)) + \log\lceil\log(1/P(m))\rceil + 5$ bits to send u (it takes $\ell + \log N$ bits).
(iv) $\log(1/Q^i_t(m)) + 5$ bits to send $h_u(m)$.

Thus, the total communication is given by,

$$\underbrace{2\log(1/Q^i_t(m))}_{(I)} + \underbrace{2\log\lceil\log(1/P(m))\rceil}_{(II)} + \underbrace{\log(\log(1/Q^i_t(m)) + 5) + 10}_{(III)}$$

We wish to prove guarantees on the expected communication cost, when m is drawn from Q. The terms in (III) are lesser order terms, which are smaller than (I), thus we choose to ignore them. Term (II) in expectation is,

$$\mathop{\mathbb{E}}_{m\sim Q}\left[\log\left(\left\lceil\log\frac{1}{P(m)}\right\rceil\right)\right] \le \log\left(\mathop{\mathbb{E}}_{m\sim Q}\left[\log\frac{1}{P(m)}\right]\right) \le \log(H(Q) + D(Q||P) + 1)$$

Term (I) is slightly more tricky to bound in expectation. Note that the empirical distribution changes on receiving message m (this turns out to be critical in bounding the communication!). That is, $Q^i_t(m) = \frac{1+(t-1)Q^i_{(t-1)}(m)}{t}$. Also let $\mathcal{M}^i_t$ be the multi-set of all messages that player i has seen up to time t. Thus, Term (I) in expectation is as follows,

$$\mathop{\mathbb{E}}_{\mathcal{M}^i_{(t-1)}}\mathop{\mathbb{E}}_{m\sim Q}\left[\log\frac{1}{Q^i_t(m)}\right] = H(Q) + \mathop{\mathbb{E}}_{\mathcal{M}^i_{(t-1)}}\mathop{\mathbb{E}}_{m\sim Q}\left[\log\frac{Q(m)}{\frac{1}{t_i} + \frac{(t_i-1)Q^i_{(t-1)}(m)}{t_i}}\right]$$

In order to bound the second term above, we consider two cases, (i) $Q^i_{(t-1)}(m) \ge Q(m)/2$ or (ii) $Q^i_{(t-1)}(m) < Q(m)/2$. After fixing t_i and m, by Chernoff bound over the randomness of $\mathcal{M}^i_{(t-1)}$ we have that case (i) happens with probability at least $1 - \exp(-t\cdot Q(m)/8)$.

$$\text{Case (i) } Q^i_{(t-1)}(m) \ge Q(m)/2 \implies \log\left(\frac{Q(m)}{\frac{1}{t_i} + \frac{(t-1)Q^i_{(t)}(m)}{t_i}}\right) \le 1$$

$$\text{Case (ii) } Q^i_{(t-1)}(m) < Q(m)/2 \implies \log\left(\frac{Q(m)}{\frac{1}{t_i} + \frac{(t_i-1)Q^i_{(t-1)}(m)}{t_i}}\right) \le \log(t_i\cdot Q(m))$$

Using these upper bounds we get that,

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathcal{M}^i_{(t-1)}}\mathop{\mathbb{E}}_{m\sim Q}\left[\log\frac{Q(m)}{\frac{1}{t} + \frac{(t_i-1)Q^i_{(t-1)}(m)}{t_i}}\right] &\le \mathop{\mathbb{E}}_{m\sim Q}\left[1\cdot\left(1 - e^{-t_i\cdot Q(m)/8}\right) + \log(t_i\cdot Q(m))\cdot e^{-t_i\cdot Q(m)/8}\right] \\ &\le 1 + \mathop{\mathbb{E}}_{m\sim Q}\left[\log(t_i\cdot Q(m))\cdot e^{-t_i\cdot Q(m)/8}\right] \\ &\le 2\,,\end{aligned}$$

where the last inequality just follows from the fact that $\log(x)\cdot e^{-x/8} \le 1$ for all x.
Thus the overall communication is bounded by

$$(2 + o(1))\,H(Q) + 2\log D(Q||P) + O(1)\,.$$

Error Guarantee

We now show that the error probability in iteration t, denoted by p_t^{err} of the protocol is upper bounded by $\frac{1}{Q(m)} \cdot 2^{-\Omega\left(\frac{t \cdot Q(m)}{K}\right)}$, where m is fixed to be the message sent in round t.

We first give an intuitive explanation for the error bound. Since player i has communicated (r, ℓ, u), player j knows the correct bucket of messages A_r to which m belongs. Knowing ℓ and u, player j also knows which hash function is being used, which is chosen to ensure that for every $m' \in S \setminus \{m\}$, $h_u(m) \neq h_u(m')$. Thus, the only way in which an error can happen is that there exists some $m' \notin S$ such that $h_u(m) = h_u(m')$ and $Q_t^j(m') > Q_t^j(m)$.

Since $m' \notin S$, it implies by definition of S that $Q_t^i(m') \leq Q_t^i(m)/16$, which means that player i has seen the message m' significantly fewer times compared to the message m. On the other hand, we also have that $Q_t^j(m') > Q_t^j(m)$, which means that player j has seen the message m' at least as many times as message m. For "large" t, it is very unlikely that players i and j have seen m and m' in such disproportionate manner.

To make the arguments go through, we need to union bound over all $m' \in A_r \setminus S$. However, a naive union bound is too lossy because we do not have any reasonable upper bound on the number of m's. To get around this issue, we do a simple bucketing argument.

The formal upper bound on p_t^{err} is shown as follows,

$$
\begin{aligned}
p_t^{\text{err}} &= \Pr[\exists m' \in A :\ h_u(m') = h_u(m) \text{ and } Q_t^j(m') > Q_t^j(m)] \\
&\leq \Pr\left[\exists m' \in A :\ Q_t^i(m') < \frac{1}{16} Q_t^i(m) \text{ and } Q_t^j(m') > Q_t^j(m)\right] \\
&\leq \Pr\left[\exists m' \in A :\ Q(m') > \frac{1}{4} Q(m) \text{ and } Q_t^i(m') < \frac{1}{16} Q_t^i(m)\right] \\
&\quad + \Pr\left[\exists m' \in A :\ Q(m') \leq \frac{1}{4} Q(m) \text{ and } Q_t^j(m') > Q_t^j(m)\right] \\
&\leq \Pr\left[\exists m' \in A :\ Q(m') > \frac{1}{4} Q(m) \text{ and } Q_{t-1}^i(m') < \frac{1}{16}\left(Q_{t-1}^i(m) + \frac{1}{t_i - 1}\right)\right] \\
&\quad + \Pr\left[\exists m' \in A :\ Q(m') \leq \frac{1}{4} Q(m) \text{ and } Q_t^j(m') > Q_t^j(m)\right] \\
&\leq \underbrace{\Pr\left[\exists m' \in A :\ Q(m') > \frac{1}{4} Q(m) \text{ and } Q_{t-1}^i(m') < \frac{1}{8} Q(m)\right]}_{(I)} \\
&\quad + \underbrace{\Pr\left[Q_{t-1}^i(m) + \frac{1}{t_i - 1} > 2 \cdot Q(m) \;\middle|\; t_i \geq \frac{t}{2K}\right]}_{(II)} + \underbrace{\Pr\left[t_i \leq \frac{t}{2K}\right]}_{(III)} \\
&\quad + \underbrace{\Pr\left[\exists m' \in A :\ Q(m') \leq \frac{1}{4} Q(m) \text{ and } Q_t^j(m') > \frac{1}{2} Q(m)\right]}_{(IV)} \\
&\quad + \underbrace{\Pr\left[Q_t^j(m) < \frac{1}{2} Q(m)\right]}_{(V)} .
\end{aligned}
$$

We bound each term individually. Firstly, since $\left\{Q_{t-1}^i(m) | t_i\right\}$ (i.e., $Q_{t-1}^i(m)$ conditioned on a fixed t_i), t_i and $Q_t^j(m)$ are binomial random variables with probabilities $Q(m)$, $\frac{1}{K}$ and $\frac{Q(m)}{K}$ respectively, the terms (II), (III) and (V) are easily upper bounded using the Chernoff

bound. In particular,

$$\begin{aligned}\Pr\left[Q^i_{t-1}(m) + \frac{1}{t_i - 1} > 2 \cdot Q(m) \;\middle|\; t_i \geq \frac{t}{2K}\right] &\leq \exp\left(-\Omega\left(\frac{t \cdot Q(m)}{K}\right)\right) \\ \Pr\left[t_i \leq \frac{t}{2K}\right] &\leq \exp\left(-\Omega\left(\frac{t \cdot Q(m)}{K}\right)\right) \\ \Pr\left[Q^j_t(m) < \frac{1}{2}Q(m)\right] &\leq \exp\left(-\Omega\left(\frac{t \cdot Q(m)}{K}\right)\right) .\end{aligned}$$

Term (I) is also upper bounded by Chernoff bound and a union bound over m', since the number of m' satisfying $Q(m') > \frac{1}{4}Q(m)$ is at most $4/Q(m)$. Thus,

$$\Pr\left[\exists m' \in A :\; Q(m') > \frac{1}{4}Q(m) \text{ and } Q^i_t(m') < \frac{1}{8}Q(m)\right] \leq \frac{4}{Q(m)} \cdot \exp(-\Omega(t_i \cdot Q(m))).$$

To bound term (IV), we can assume without loss of generality that there is at most one $m' \in A$, such that, $Q(m') \leq \frac{1}{8}Q(m)$. This is because, if there were to exist $m'_1, m'_2 \in A$, such that, $Q(m'_1), Q(m'_2) \leq \frac{1}{8}Q(m)$, then we can identify m'_1 and m'_2 as the same message m'_0. Note that we can do this because we will still have that $\Pr[Q(m'_0) \leq \frac{1}{4}Q(m)]$ and

$$\Pr\left[Q^j_t(m'_1) > \frac{1}{2}Q(m) \text{ or } Q^j_t(m'_2) > \frac{1}{2}Q(m)\right] \leq \Pr\left[Q^j_t(m'_0) > \frac{1}{2}Q(m)\right]$$

Thus, to bound term (IV), we can again use a Chernoff bound and a union bound over m', since the number of m' such that $Q(m') > \frac{1}{8}Q(m)$ is at most $8/Q(m)$. Thus, we get that,

$$\Pr\left[\exists m' \in A :\; Q(m') \leq \frac{1}{4}Q(m) \text{ and } Q^j_t(m') > \frac{1}{2}Q(m)\right] \leq \frac{1}{Q(m)} \cdot \exp\left(-\Omega\left(\frac{t \cdot Q(m)}{K}\right)\right)$$

Thus, overall in any individual round, we have that,

$$p^{\text{err}}_t \leq \frac{1}{Q(m)} \cdot \exp\left(-\Omega\left(\frac{t \cdot Q(m)}{K}\right)\right).$$

This concludes the proof of Lemma 5.

Outlaw Distributions and Locally Decodable Codes

Jop Briët[*1], **Zeev Dvir**[†2], **and Sivakanth Gopi**[‡3]

1 **CWI, Amsterdam, The Netherlands**
j.briet@cwi.nl
2 **Dept. of Computer Science and Dept. of Mathematics, Princeton University, Princeton, USA**
zeev.dvir@gmail.com
3 **Dept. of Computer Science, Princeton University, Princeton, USA**
sgopi@cs.princeton.edu

Abstract

Locally decodable codes (LDCs) are error correcting codes that allow for decoding of a single message bit using a small number of queries to a corrupted encoding. Despite decades of study, the optimal trade-off between query complexity and codeword length is far from understood. In this work, we give a new characterization of LDCs using distributions over Boolean functions whose expectation is hard to approximate (in L_∞ norm) with a small number of samples. We coin the term 'outlaw distributions' for such distributions since they 'defy' the Law of Large Numbers. We show that the existence of outlaw distributions over sufficiently 'smooth' functions implies the existence of constant query LDCs and vice versa. We give several candidates for outlaw distributions over smooth functions coming from finite field incidence geometry and from hypergraph (non)expanders.

We also prove a useful lemma showing that (smooth) LDCs which are only required to work on average over a random message and a random message index can be turned into true LDCs at the cost of only constant factors in the parameters.

1998 ACM Subject Classification E.4 Coding and Information Theory

Keywords and phrases Locally Decodable Code, VC-dimension, Incidence Geometry, Cayley Hypergraphs

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.20

1 Introduction

Error correcting codes (ECCs) solve the basic problem of communication over noisy channels. They encode a message into a codeword from which, even if the channel partially corrupts it, the message can later be retrieved. With one of the earliest applications of the *probabilistic method*, formally introduced by Erdős in 1947, pioneering work of Shannon [25] showed the existence of optimal (capacity-achieving) ECCs. The problem of explicitly constructing such codes has fueled the development of coding theory ever since. Similarly, the exploration of many other fascinating structures, such as Ramsey graphs, expander graphs, two source

* The author supported by a VENI grant and the Gravitation-grant NETWORKS-024.002.003 from the Netherlands Organisation for Scientific Research (NWO).
† The author supported by NSF grant CCF-1523816 and by the Sloan foundation.
‡ The author supported by NSF grant CCF-1523816 and by the Sloan foundation.

© Jop Briët, Zeev Dvir and Sivakanth Gopi;
licensed under Creative Commons License CC-BY

extractors, etc., began with a striking existence proof via the probabilistic method, only to be followed by decades of catch-up work on explicit constructions. Locally decodable codes (LDCs) are a special class of error correcting codes whose development has not followed this line. The defining feature of LDCs is that they allow for ultra fast decoding of single message bits, a property that typical ECCs lack, as their decoders must read an entire (possibly corrupted) codeword to achieve the same. They were first formally defined in the context of channel coding in [18], although they (and the closely related locally correctable codes) implicitly appeared in several previous works in other settings, such as program checking [5], probabilistically checkable proofs [4, 3] and private information retrieval schemes (PIRs) [9]. More recently, LDCs have even found applications in Banach-space geometry [7] and LDC-inspired objects called local reconstruction codes found applications in fault tolerant distributed storage systems [15]. See [34] for a survey of LDCs and some of the applications.

Despite their many applications, our knowledge of LDCs is very limited; the best-known constructions are far from what is currently known about their limits. Although standard random (linear) ECCs do allow for some weak local-decodability, they are outperformed by even the earliest explicit constructions [19]. All the known constructions of LDCs were obtained by explicitly designing such codes using some algebraic objects like low-degree polynomials or matching vectors [34].

In this paper, we give a characterization of LDCs in probabilistic and geometric terms, making them amenable to probabilistic constructions. On the flip side, these characterizations might also be easier to work with for the purpose of showing lower bounds. We will make this precise in the next section. Let us first give the formal definition of an LDC.

▶ **Definition 1** (Locally decodable code)**.** For positive integers k, n, q and $\eta, \delta \in (0, 1/2]$, a map $C : \{0,1\}^k \to \{0,1\}^n$ is a (q, δ, η)-*locally decodable code* if, for every $i \in [k]$, there exists a randomized *decoder* (a probabilistic algorithm) $\mathcal{A}_i$ such that:

- For every message $x \in \{0,1\}^k$ and string $y \in \{0,1\}^n$ that differs from the codeword $C(x)$ in at most δn coordinates,

$$\Pr[\mathcal{A}_i(y) = x_i] \geq \frac{1}{2} + \eta. \tag{1}$$

- The decoder $\mathcal{A}_i$ (non-adaptively) queries at most q coordinates of y.[1]

Known results

The main parameters of LDCs are the number of queries q and the length of the encoding n as a function of k and q, typically the parameters δ, η are some fixed constants. The simplest example is the Hadamard code, which is a 2-query LDC with $n = 2^k$. The 2-query regime is the only nontrivial case where optimal lower bounds are known: it was shown in [20, 14] that exponential length is necessary. In general, Reed-Muller codes of degree $q - 1$ are q-query LDCs of length $n = \exp(O(k^{1/q-1}))$. For a long time, these were the best constructions for constant q, until in a breakthrough work by [33, 13], 3-query LDCs were constructed with subexponential length $n = \exp(\exp(O(\sqrt{\log k})))$. More generally they constructed 2^r-query LDCs with length $n = \exp(\exp(O(\log^{1/r} k)))$. For $q \geq 3$, the best-known lower bounds leave huge gaps, giving only polynomial bounds. Any 3-query

[1] We can assume that on input $y \in \{0,1\}^n$, the decoder $\mathcal{A}_i$ first samples a set $S \subseteq [n]$ of at most q coordinates according to a probability distribution depending on i only and then returns a random bit depending only on i, S and the values of y at S.

LDC must have length $n \geq \tilde{\Omega}(k^2)$ [32], and more generally any q-query LDC must have length $n \geq \tilde{\Omega}(k^{1+1/(\lceil q/2 \rceil - 1)})$[18, 31]. LDCs where the codewords are over a large alphabet are also studied because of their relation to private information retrieval schemes [9, 18]. In [12], 2-query LDCs of length $n = \exp(k^{o(1)})$ over an alphabet of size $\exp(k^{o(1)})$ were constructed. There is also some exciting recent work on LDCs when the number of queries can also grow with k, in which case there are explicit constructions with constant-rate (that is, $n = O(k)$) and query complexity $q = \exp(O(\sqrt{\log n}))$; in fact we can even achieve the optimal rate-distance tradeoff of traditional error correcting codes [23, 22, 16]. We cannot yet rule out the exciting possibility that constant rate LDCs with polylogarithmic query complexity exist.

1.1 LDCs from distributions over smooth Boolean functions

Our main result shows that LDCs can be obtained from "outlaw" distributions over "smooth" functions. The term outlaw refers to the Law of Large Numbers, which says that the average of independent samples tends to the expectation of the distribution from which they are drawn. Roughly speaking, a probability distribution is an outlaw if many samples are needed for a good estimation of the expectation and a smooth function over the n-dimensional Boolean hypercube is one that has no influential variables. Paradoxically, while many instances of the probabilistic method use the fact that sample means of a small number of independent random variables tend to concentrate around the true mean, as captured for example by the Chernoff bound, our main result requires precisely the opposite. We show that if *at least* k samples from a distribution over smooth functions are needed to approximate the mean, then there exists an $O(1)$-query LDC sending $\{0,1\}^{\Omega(k)}$ to $\{0,1\}^n$, where the hidden constants depend only the smoothness and mean-estimation parameters.

To make this precise, we now formally define smooth functions and outlaw distributions. Given a function $f : \{-1,1\}^n \to \mathbb{R}$, its *spectral norm* (also known as the algebra norm or Wiener norm) is defined as

$$\|f\|_{\mathrm{sp}} = \sum_{S \subset [n]} |\widehat{f}(S)|,$$

where $\widehat{f}(S)$ are the Fourier coefficients of f (see Section 2 for some preliminaries in Fourier analysis). We also consider the supremum norm, $\|f\|_{L_\infty} = \sup\{|f(x)| : x \in \{-1,1\}^n\}$. It follows from the Fourier inversion formula that $\|f\|_{L_\infty} \leq \|f\|_{\mathrm{sp}}$. The *ith discrete derivative* of f is the function $(D_i f)(x) = (f(x) - f(x^i))/2$, where x^i is the point that differs from x on the ith coordinate. Smooth functions are functions whose discrete derivatives have small spectral norms.

▶ **Definition 2** (σ-smooth functions)**.** For $\sigma > 0$, a function $f : \{-1,1\}^n \to \mathbb{R}$ is σ-smooth, if for every $i \in [n]$, we have $\|D_i f\|_{\mathrm{sp}} \leq \sigma/n$.

Intuition for the above definition may be gained from the fact that smooth functions have no influential variables. The influence of the ith variable, $(\mathbb{E}_{x \in \{-1,1\}^n}[(D_i f)(x)^2])^{1/2}$, measures the extent to which changing the ith coordinate of a randomly chosen point changes the value of f. Since $\|D_i f\|_{L_\infty} \leq \|D_i f\|_{\mathrm{sp}}$, the directional derivatives of σ-smooth functions are uniformly bounded by σ/n, which is a much stronger condition than saying that the derivatives are small on average. Outlaws are defined as follows.

▶ **Definition 3** (Outlaw)**.** Let n be a positive integer and μ be a probability distribution over real-valued functions on $\{-1,1\}^n$. For a positive integer k and $\varepsilon > 0$, say that μ is a

(k, ε)-*outlaw* if for independent random μ-distributed functions $f_1, \ldots, f_k$ and $\bar{f} = \mathbb{E}_\mu[f]$,

$$\mathbb{E}\left[\left\|\frac{1}{k}\sum_{i=1}^{k}(f_i - \bar{f})\right\|_{L_\infty}\right] \geq \varepsilon.$$

Denote by $\kappa_\mu(\epsilon)$ the largest integer k such that μ is a (k, ε)-outlaw.

To approximate the true mean of an outlaw μ to within ε on average in the L_∞-distance, one thus needs $\kappa_\mu(\varepsilon)+1$ samples. Note that if μ is a distribution over σ-smooth functions, then the distribution $\tilde{\mu}$ obtained by scaling functions in the support of μ by $1/\sigma$ is a distribution over 1-smooth functions and $\kappa_{\tilde{\mu}}(\varepsilon/\sigma) = \kappa_\mu(\varepsilon)$.

Our main result is then as follows.

▶ **Theorem 4** (Main theorem)**.** *Let n be a positive integer and $\varepsilon > 0$. Let μ be a probability distribution over 1-smooth functions on $\{-1, 1\}^n$ and $k = \kappa_\mu(\varepsilon)$. Then, there exists a (q, δ, η)-LDC sending $\{0, 1\}^l$ to $\{0, 1\}^n$ where $l = \Omega(\varepsilon^2 k/\log(1/\varepsilon))$, $q = O(1/\varepsilon)$, $\delta = \Omega(\varepsilon)$ and $\eta = \Omega(\varepsilon)$. Additionally, if μ is supported by degree-d functions, then we can take $q = d$.*

Note that the smoothness requirement is essential. For example the uniform distribution over the n dictator functions $f_i(x) = x_i$ for $i \in [n]$ is an $(n/2, 1)$-outlaw, but it cannot imply constant rate, constant query LDCs which we know do not exist. In fact we establish a converse to Theorem 4, showing that its hypothesis is essentially equivalent to the existence of LDCs in the small query complexity regime.

▶ **Theorem 5.** *If $C : \{0, 1\}^k \to \{0, 1\}^n$ is a (q, δ, η)-LDC, then there exists a probability distribution μ over 1-smooth degree-q functions on $\{-1, 1\}^n$ such that*

$$\kappa_\mu(\varepsilon) \geq \eta k$$

where $\varepsilon = \eta\delta/(q2^{q/2})$.

Let us remark in passing that Theorem 5 can in turn convert the problem of proving lower bounds on the length of LDCs to a problem on Banach space geometry. In particular, it can be shown that for a distribution μ over 1-smooth degree-q functions on $\{0, 1\}^n$, one can upper bound $\kappa_\mu(\varepsilon)$ in terms of type constants of the space of q-linear forms on ℓ_q^{n+1} [6].

Candidate outlaws

One scenario in which outlaw distributions can be obtained is using incidence geometry in finite fields. In particular, the following result can be derived from our main theorem (stated a bit informally here, see Section 6.1 for the formal version).

▶ **Corollary 6.** *Let $p > 2$ be a fixed prime. Suppose that for every set of directions $D \subset \mathbb{F}_p^n$ of size $|D| \leq k$, there exists a set $B \subset \mathbb{F}_p^n$ of size $|B| \geq \Omega(p^n)$ which does not contain any lines with direction in D. Then, there exists a p-query LDC sending $\{0, 1\}^{\Omega(k)}$ to $\{0, 1\}^{p^n}$.*

Another setting in which our approach leads to interesting open problems is in relation to expansion in hypergraphs. Consider a partition of the complete bipartite graph $K_{n,n}$ into n perfect matchings. It is known that picking $k = O(\log(n))$ of these matchings at random will give us an expander graph (of degree k). For some particular partitions (e.g., given by an Abelian group) this bound is tight. The questions arising from our approach can be briefly summarized as follows: Can one find an n-vertex hypergraph H (say three uniform to be precise) and a partition of H into matchings so that, to get an expander (defined appropriately) one needs at least k random matchings. This would give a code sending k bit messages with encoding length n and so, becomes interesting when k is super poly-logarithmic in n. We elaborate on this in Section 6.2

1.2 Techniques

Our proof of Theorem 4 proceeds in two steps. The first step consists of turning an outlaw over smooth functions into a seemingly crude type of LDC that is only required to work on average over a uniformly distributed message and a uniformly distributed message index. We call such codes *average-case smooth codes* (see below). The second step consists of showing that such codes are in fact not much weaker than honest LDCs.

From outlaws to average-case smooth codes

The key ingredient for the first step is *symmetrization*, a basic technique from high-dimensional probability. We briefly sketch how this is used (we refer to Section 3 for the full proof). Suppose that $f_1, \ldots, f_k$ are independent smooth functions distributed according to a (k, ε)-outlaw with expectation $\bar{f}$. We introduce an independent copy[2] f_i' of f_i for each $i \in [k]$ and consider the symmetrically distributed random functions $f_i - f_i'$. Since $\bar{f} = \mathbb{E}[f_i']$ for each $i \in [k]$, Jensen's inequality and Definition 3 imply that

$$\mathbb{E}\big[\|(f_1 - f_1') + \cdots + (f_k - f_k')\|_{L_\infty}\big] \geq \mathbb{E}\big[\|(f_1 - \mathbb{E}[f_1']) + \cdots + (f_k - \mathbb{E}[f_k')\|_{L_\infty}\big] \geq \varepsilon k.$$

Since the random functions $f_i - f_i'$ are independent and symmetric, we get that for independent uniformly random signs $x_1, \ldots, x_k \in \{-1, 1\}$, the above left-hand side equals

$$\mathbb{E}\big[\|x_1(f_1 - f_1') + \cdots + x_k(f_k - f_k')\|_{L_\infty}\big].$$

The triangle inequality and the Averaging Principle then give that there exist *fixed* smooth functions $f_1^\star, \ldots, f_k^\star$ such that on average over the random signs, we have

$$\mathbb{E}\big[\|x_1 f_1^\star + \cdots + x_k f_k^\star\|_{L_\infty}\big] \geq \varepsilon k/2. \tag{2}$$

To get an average-case smooth code out of this, we view each sequence $x = (x_1, \ldots, x_k)$ as a k-bit message and choose an arbitrary n-bit string for which the L_∞-norm in (2) is achieved to be the its encoding, $C(x)$. This gives a map $C : \{-1, 1\}^k \to \{0, 1\}^n$ satisfying

$$\mathbb{E}\big[x_1 f_1^\star(C(x)) + \cdots x_k f_k^\star(C(x))\big] \geq \varepsilon k/2.$$

Equivalently, for uniform x and i, we have $\Pr[f_i^\star(C(x)) = x_i] \geq \frac{1}{2} + \frac{\varepsilon}{4}$. Finally, we use the smoothness property to transform the $f_i^\star$ into decoders with the desired properties. This is done in Section 3. Let us point out that it is in the application of the Averaging Principle where the probabilistic method appears in our construction of LDCs.

Average-case smooth codes are LDCs

Katz and Trevisan [18] observed that LDC decoders must have the property that they select their queries according to distributions that do not favor any particular coordinate. The intuition for this is that if they did favor a certain coordinate, then corrupting that coordinate would cause the decoder to err with too high a probability. If instead, queries are sampled according to a "smooth" distribution, they will all fall on uncorrupted coordinates with good probability provided the fraction of corrupted coordinates δ and query complexity q aren't too large. The following definition allows us to make this intuition precise.

[2] in this context sometimes referred to as a "ghost copy" as it will later disappear again

▶ **Definition 7** (Smooth code)**.** For positive integers k, n, q and parameters $\eta \in (0, 1/2]$ and $c > 0$, a map $C : \{0,1\}^k \to \{0,1\}^n$ is a (q, c, η)-*smooth code* if, for every $i \in [k]$, there exists a randomized decoder $\mathcal{A}_i : \{0,1\}^n \to \{0,1\}$ such that

- For every $x \in \{0,1\}^k$,

$$\Pr\big[x_i = \mathcal{A}_i\big(C(x)\big)\big] \geq \frac{1}{2} + \eta. \tag{3}$$

- The decoder $\mathcal{A}_i$ (non-adaptively) queries at most q coordinates of $C(x)$.
- For each $j \in [n]$, the probability that $\mathcal{A}_i$ queries the coordinate $j \in [n]$ is at most c/n.

The formal version of Katz and Trevisan's observation is as follows.

▶ **Theorem 8** (Katz–Trevisan)**.** *If $C : \{0,1\}^k \to \{0,1\}^n$ is a (q, δ, η)-LDC, then C is also a $(q, q/\delta, \eta)$-smooth code. Conversely, if $C : \{0,1\}^k \to \{0,1\}^n$ is a (q, c, η)-smooth code, then C is also a $(q, \eta/2c, \eta/2)$-LDC.*

Our second step in the proof of Theorem 4 is a stronger form of the converse part of Theorem 8. We show that even smooth codes that are only required to work *on average* can be turned into LDCs, losing only a constant factor in the rate and success probability.

▶ **Definition 9** (Average-case smooth code)**.** A code as in Definition 7 is a (q, c, η)-*average-case* smooth code if instead of the first item, (3) is required to hold only on average over uniformly distributed $x \in \{0,1\}^k$ and uniformly distributed $i \in [k]$, which is to say that

$$\Pr\big[x_i = \mathcal{A}_i\big(C(x)\big)\big] \geq \frac{1}{2} + \eta,$$

where the probability is taken over x, i and the randomness used by $\mathcal{A}_i$.

▶ **Lemma 10.** *Let $C : \{0,1\}^k \to \{0,1\}^n$ be a (q, c, η)-average-case smooth code. Then, there exists an $(q, \Omega(\eta/c), \Omega(\eta))$-LDC sending $\{0,1\}^l$ to $\{0,1\}^n$ where $l = \Omega(\eta^2 k/\log(1/\eta))$.*

The idea behind the proof of Lemma 10 is as follows. We first switch the message and codeword alphabets to $\{-1, 1\}$ and let $f_i : \{-1,1\}^n \to [-1,1]$ be the expected decoding function $f_i(z) = \mathbb{E}[\mathcal{A}_i(z)]$. The properties of C then easily imply that the set $T \subseteq [-1,1]^k$ given by $T = \{(f_1(z), \ldots, f_k(z)) : z \in \{-1,1\}^n\}$ has large *Gaussian width*, in particular it holds that for a standard k-dimensional Gaussian vector g, we have $\mathbb{E}[\sup_{t \in T} \langle g, t\rangle] \gtrsim \varepsilon k$.[3] Next, we employ a powerful result of [24] showing that T contains an l-dimensional hypercube-like structure with edge length some absolute constant $c \in (0, 1]$, for $l \gtrsim k$. Roughly speaking, this implies that C is a smooth code on $\{-1,1\}^l$ whose decoding probability depends on ε and c. Finally, we obtain an LDC via an application of Theorem 8. The full proof is given in Section 4.

1.3 Organization

Section 2 contains some preliminaries in Fourier analysis over the Boolean cube. In Section 3, we prove our main theorem (Theorem 4) by first showing that outlaw distributions over smooth functions imply existence of average-case smooth codes and using Lemma 10 to convert them to LDCs. In Section 4, we prove Lemma 10 showing how to convert average-case

[3] We write $A \gtrsim B$ and $A = \Omega(B)$ interchangeably to mean that $A \geq cB$ for some absoulte constant $c > 0$ independent of all parameters involved.

smooth-codes to LDCs. In Section 5, we show the converse to our main theorem (Theorem 5) showing how to get outlaw distributions over smooth functions from LDCs. Finally in Section 6, we give some candidate constructions of outlaw distributions over smooth functions using incidence geometry and Cayley hypergraphs.

2 Preliminaries

We recall a few basic definitions and facts from analysis over the n-dimensional Boolean hypercube $\{-1,1\}^n$. Equipped with the coordinate-wise multiplication operation, the hypercube forms an Abelian group whose group of characters is formed by the functions $\chi_S(x) = \prod_{i\in S} x_i$ for all $S \subseteq [n]$. The characters form a complete orthonormal basis for the space of real-valued functions on $\{-1,1\}^n$ endowed with the inner product $\langle f, g\rangle = \mathbb{E}_{x\in\{-1,1\}^n}[f(x)g(x)]$, where we use the notation $\mathbb{E}_{a\in S}$ to denote the expectation with respect to a uniformly distributed element a over a set S. The *Fourier transform* of a function $f : \{-1,1\}^n \to \mathbb{R}$ is the function $\widehat{f} : 2^{[n]} \to \mathbb{R}$ defined by $\widehat{f}(S) = \langle f, \chi_S\rangle$. The Fourier inversion formula (which follows from orthonormality of the character functions) asserts that

$$f = \sum_{S\subseteq[n]} \widehat{f}(S)\chi_S.$$

Parseval's Identity relates the L_2-norms of f and its Fourier transform by

$$\left(\mathbb{E}_{x\in\{-1,1\}^n}[f(x)^2]\right)^{1/2} = \Big(\sum_{S\subseteq[n]} |\widehat{f}(S)|^2\Big)^{1/2}.$$

A function f has *degree* q if $\widehat{f}(S) = 0$ when $|S| > q$ and the *degree-q truncation* of f, denoted $f^{\leq q}$, is the degree-q function defined by

$$f^{\leq q} = \sum_{|S|\leq q} \widehat{f}(S)\chi_S.$$

A function f is a *q-junta* if it depends only on a subset of q of its variables, or equivalently, if there exists a subset $T \subseteq [n]$ of size $|T| \leq q$ such that $\widehat{f}(S) = 0$ for every $S \not\subseteq T$. The *ith discrete derivative* $D_i f$ is the function $(D_i f)(x) = (f(x) - f(x^i))/2$, where x^i is the point that differs from x on the ith coordinate. It is easy to show that the ith discrete derivative in of a function f is given by

$$D_i f = \sum_{S\ni i} \widehat{f}(S)\chi_S.$$

Hence, it follows that $\|D_i f\|_{\mathrm{Spec}} = \sum_{S\ni i} |\widehat{f}(S)|$.

3 From outlaws to LDCs

In this section we prove Theorem 4. For convenience, in the remainder of this paper, we switch the message and codeword alphabets of all codes from $\{0,1\}^n$ to $\{-1,1\}^n$. We begin by showing that outlaw distributions over degree-q functions give q-query average-case smooth codes. Combined with Lemma 10, this implies the second part of Theorem 4.

▶ **Theorem 11.** *Let μ be a probability distribution on 1-smooth degree-q functions on $\{-1,1\}^n$, let $\varepsilon \in (0,1]$ and let $k = \kappa_\mu(\varepsilon)$. Then, there exists a $(q, 1, \varepsilon/4)$-average-case smooth code sending $\{-1,1\}^k$ to $\{-1,1\}^n$.*

Proof. The proof uses a symmetrization argument. Let $\mathcal{F} = (f_1, \ldots, f_k), \mathcal{F}' = (f'_1, \ldots, f'_k)$ be two k-tuples of independent μ-distributed random variables and let $\bar{f} = \mathbb{E}_\mu[f]$. Then, by definition of $\kappa_\mu(\varepsilon)$ and Jensen's inequality,

$$\begin{aligned}
\varepsilon &\leq \mathbb{E}_{\mathcal{F}}\Big[\Big\|\frac{1}{k}\sum_{i=1}^{k}(f_i - \bar{f})\Big\|_{L_\infty}\Big] \\
&= \mathbb{E}_{\mathcal{F}}\Big[\Big\|\frac{1}{k}\sum_{i=1}^{k}\big(f_i - \mathbb{E}_{\mathcal{F}'}[f'_i]\big)\Big\|_{L_\infty}\Big] \\
&\leq \mathbb{E}_{\mathcal{F},\mathcal{F}'}\Big[\Big\|\frac{1}{k}\sum_{i=1}^{k}(f_i - f'_i)\Big\|_{L_\infty}\Big].
\end{aligned}$$

The random variables $f_i - f'_i$ are symmetrically distributed, which is to say that they have the same distribution as their negations $f'_i - f_i$. Since they are independent, it follows that for every $x \in \{-1,1\}^k$, the random variable $x_1(f_1 - f'_1) + \cdots + x_k(f_k - f'_k)$ has the same distribution as $(f_1 - f'_1) + \cdots + (f_k - f'_k)$. Therefore,

$$\begin{aligned}
\mathbb{E}_{\mathcal{F},\mathcal{F}'}\Big[\Big\|\frac{1}{k}\sum_{i=1}^{k}(f_i - f'_i)\Big\|_{L_\infty}\Big] &= \mathbb{E}_{x\in\{-1,1\}^k}\Big[\mathbb{E}_{\mathcal{F},\mathcal{F}'}\Big[\Big\|\frac{1}{k}\sum_{i=1}^{k}x_i(f_i - f'_i)\Big\|_{L_\infty}\Big]\Big] \\
&\leq 2\mathbb{E}_{\mathcal{F}}\Big[\mathbb{E}_{x\in\{-1,1\}^k}\Big[\Big\|\frac{1}{k}\sum_{i=1}^{k}x_i f_i\Big\|_{L_\infty}\Big]\Big].
\end{aligned}$$

Applying the Averaging Principle to the outer expectation, we find that there exist 1-smooth degree-q functions $f_1^\star, \ldots, f_k^\star : \{-1,1\}^n \to \mathbb{R}$ such that

$$\mathbb{E}_{x\in\{-1,1\}^k}\Big[\Big\|\frac{1}{k}\sum_{i=1}^{k}x_i f_i^\star\Big\|_{L_\infty}\Big] \geq \frac{\varepsilon}{2}. \tag{4}$$

Define the code $C : \{-1,1\}^k \to \{-1,1\}^n$ such that for each $x \in \{-1,1\}^k$, we have

$$\frac{1}{k}\sum_{i=1}^{k}x_i f_i^\star(C(x)) = \Big\|\frac{1}{k}\sum_{i=1}^{k}x_i f_i^\star\Big\|_{L_\infty}. \tag{5}$$

For each $i \in [k]$, define the decoder $\mathcal{A}_i$ as follows. Let $\nu_i : 2^{[n]} \to [0,1]$ be the probability distribution defined by $\nu_i(S) = |\widehat{f_i^\star}(S)|/\|\widehat{f_i^\star}\|_{\mathrm{sp}}$. Given a string $z \in \{-1,1\}^n$, with probability $1 - \|\widehat{f_i^\star}\|_{\mathrm{sp}}$, the decoder $\mathcal{A}_i$ returns a uniformly random sign, and with probability $\|\widehat{f_i^\star}\|_{\mathrm{sp}}$, it samples a set $S \subseteq [n]$ according to ν_i and returns $\chi_S(z)$. This is a valid probability distribution since for any 1-smooth function f, we have

$$\|f\|_{\mathrm{sp}} = \sum_{S\subset[n]}|\widehat{f}(S)| \leq \sum_{S\subset[n]}|S||\widehat{f}(S)| = \sum_{i=1}^{n}\sum_{S\ni i}|\widehat{f}(S)| \leq n\cdot\frac{1}{n} = 1.$$

Then, $\mathcal{A}_i$ queries at most q coordinates of z and since $f_i^\star$ is 1-smooth, the probability that it queries any coordinate $j \in [n]$ is at most $\|D_j f_i^\star\|_{\mathrm{sp}} \leq 1/n$.

We also have $\mathbb{E}[\mathcal{A}_i(z)] = f_i^\star(z)$. Therefore, by (4) and (5), we have

$$\begin{aligned}\mathbb{E}_{x\in\{-1,1\}^k, i\in[k]}\left[\Pr[x_i = \mathcal{A}_i(C(x))]\right] &= \frac{1}{2} + \frac{1}{2}\mathbb{E}_{x\in\{-1,1\}^k, i\in[k]}\left[x_i\mathbb{E}[\mathcal{A}_i(C(x))]\right] \\ &= \frac{1}{2} + \frac{1}{2}\mathbb{E}_{x\in\{-1,1\}^k, i\in[k]}\left[x_i f_i^\star(C(x))\right] \\ &= \frac{1}{2} + \frac{1}{2}\mathbb{E}_{x\in\{-1,1\}^k}\left[\left\|\frac{1}{k}\sum_{i=1}^k x_i f_i^\star\right\|_{L_\infty}\right] \\ &\geq \frac{1}{2} + \frac{\varepsilon}{4}.\end{aligned}$$

Hence, C is a $(q, 1, \varepsilon/4)$-average-case smooth code. ◀

The final step before the proof of Theorem 4 is to show that for any distribution μ over smooth functions, there exists a distribution $\tilde{\mu}$ over smooth functions of bounded degree that is not much more concentrated than μ.

▶ **Lemma 12.** *Let μ be a probability distribution over 1-smooth functions on $\{-1,1\}^n$ and let $\varepsilon > 0$. Then, there exists a probability distribution $\tilde{\mu}$ over 1-smooth functions of degree $q = 4/\epsilon$ such that $\kappa_{\tilde{\mu}}(\epsilon/2) \geq \kappa_\mu(\epsilon)$.*

Proof. We first establish that smooth functions have low-degree approximations in the supremum norm. If $f : \{-1,1\}^n \to \mathbb{R}$ is 1-smooth, then

$$q\sum_{|S|>q}|\widehat{f}(S)| \leq \sum_{S\subset[n]}|S||\widehat{f}(S)| = \sum_{i=1}^n\sum_{S\ni i}|\widehat{f}(S)| = \sum_{i=1}^n\|D_i f\|_{\mathrm{sp}} \leq 1.$$

It follows that the degree-q truncation $f^{\leq q}$ satisfies

$$\left\|f - f^{\leq q}\right\|_{L_\infty} \leq \sum_{|S|>q}|\widehat{f}(S)| \leq \frac{1}{q} = \frac{\epsilon}{4}. \tag{6}$$

Define $\tilde{\mu}$ as follows: sample f according to μ and output $f^{\leq q}$. Clearly, $\tilde{\mu}$ is also a distribution over 1-smooth functions. For $k = \kappa_\mu(\varepsilon)$, we have

$$\mathbb{E}_{f_1,\ldots,f_k\sim\mu}\left[\left\|\frac{1}{k}\sum_{i=1}^k\left(f_i - \mathbb{E}[f_i]\right)\right\|_{L_\infty}\right] \geq \varepsilon.$$

Hence, by the triangle inequality and (6), we have

$$\mathbb{E}_{f_1,\ldots,f_k\sim\tilde{\mu}}\left[\left\|\frac{1}{k}\sum_{i=1}^k\left(f_i - \mathbb{E}[f_i]\right)\right\|_{L_\infty}\right] \geq \frac{\epsilon}{2},$$

giving the claim. ◀

Proof of Theorem 4. By applying Lemma 12 to μ, we get a distribution $\tilde{\mu}$ over 1-smooth degree $q = O(1/\epsilon)$ functions with $k' = \kappa_{\tilde{\mu}}(\epsilon/2) \geq \kappa_\mu(\epsilon) = k$. By Theorem 11, we get a $(q, 1, \Omega(\epsilon))$-average-case smooth code $C' : \{-1,1\}^{k'} \to \{-1,1\}^n$. Finally we use Lemma 10 to convert C' to a $(q, \Omega(\epsilon), \Omega(\epsilon))$-LDC $C : \{-1,1\}^\ell \to \{-1,1\}^n$ where $\ell = \Omega(\epsilon^2 k'/\log(1/\epsilon))$. For the last part of the theorem we can simply apply Theorem 11 directly. ◀

4 From average-case smooth codes to LDCs

In this section, we prove Lemma 10. For this, we need the notion of the Vapnik–Chervonenkis dimension (VC-dimension).

▶ **Definition 13** (VC-dimension)**.** For $T \subset [-1,1]^k$ and $w > 0$, $\mathrm{vc}(T,w)$ is defined as the size of the largest subset $\sigma \subset [k]$ such that there exists a shift $s \in [-1,1]^k$ satisfying the following: for every $x \in \{-1,1\}^\sigma$, there exists $t \in T$ such that for every $i \in \sigma$, $(t_i - s_i)x_i \geq w/2$.

Observe that if T is convex, then $\mathrm{vc}(T,w)$ is the maximum dimension of a shifted hypercube with edge lengths at least w contained in T .

▶ **Definition 14** (Gaussian width)**.** Let g be a k-dimensional standard Gaussian vector, with independent standard normal distributed entries. The *Gaussian width* of a set $T \subseteq \mathbb{R}^k$ is defined as

$$E(T) = \mathbb{E}_g[\sup_{t\in T}\langle g,t\rangle].$$

It is easy to see that a large VC-dimension implies a large Gaussian width. The following theorem shows the converse: containing a hypercube-like structure is the only way to have large Gaussian width.

▶ **Theorem 15** ([24])**.** *Let $T \subset [-1,1]^k$. Then, the Gaussian width of T is bounded as*

$$E(T) \lesssim \sqrt{k}\int_{\alpha E(T)/k}^{1} \sqrt{\mathrm{vc}(T,w)\log(1/w)}dw$$

for some absolute constant $\alpha > 0$.

Finally, we use that fact that, as for LDCs, we can assume that on input $y \in \{0,1\}^n$, the decoder $\mathcal{A}_i$ of a smooth code first samples a set $S \subseteq [n]$ of at most q coordinates according to a probability distribution that depends on i only and then returns a random sign depending only on i, S and the values of y at S.

Proof of Lemma 10. The proof works by showing that the average-case smooth code property implies that the image of the (average) decoding functions should have large Gaussian width. We then use Theorem 15 to find a hypercube like structure inside the image, which we use to construct a smooth code. Finally we use Theorem 8 to convert the smooth code to an LDC.

Recall the switch of the message and codeword alphabets to $\{-1,1\}$. For each $i \in [k]$, let $f_i : \{-1,1\}^n \to [-1,1]$ be the expected decoding function $f_i(z) = \mathbb{E}[\mathcal{A}_i(z)]$. Let g be a standard k-dimensional Gaussian vector and $T = \{(f_1(z),\ldots,f_k(z)) : z \in \{-1,1\}^n\}$. By the definition of average-case smooth code we have

$$2\eta k \leq \mathbb{E}_{x\in\{-1,1\}^k}\left[\sum_{i=1}^{k} x_i f_i(C(x))\right] \leq \mathbb{E}_{x\in\{-1,1\}^k}\left[\sup_{t\in T}\langle x,t\rangle\right] \lesssim \mathbb{E}_g\left[\sup_{t\in T}\langle g,t\rangle\right].$$

(See for instance [26, Lemma 3.2.10] for the last inequality.) By Theorem 15, for some constant $\alpha > 0$, we have

$$\eta k \lesssim \sqrt{k}\int_{\alpha\eta}^{1}\sqrt{\mathrm{vc}(T,w)\log(1/w)}dt \leq \sqrt{k}\cdot\sqrt{\mathrm{vc}(T,\alpha\eta)\log(1/\alpha\eta)}$$

where we used the fact that $\mathrm{vc}(T, w)$ is decreasing in w. So for $\tau = \alpha\eta$, we have $\mathrm{vc}(T, \tau) \gtrsim \eta^2 k/\log(1/\eta)$. By the definition of VC-dimension, there exists a subset $\sigma \subset [k]$ of size $|\sigma| \geq \mathrm{vc}(T, \tau)$ and a shift $s \in [-1, 1]^k$ such that for every $x \in \{-1, 1\}^\sigma$ there exists $t \in T$ such that $(t_i - s_i)x_i \geq \tau/2$ for every $i \in \sigma$.

Now we will define the code $C' : \{-1, 1\}^\sigma \to \{-1, 1\}^n$. Given $x \in \{-1, 1\}^\sigma$, there exists $t(x) \in T$ such that $(t(x)_i - s_i)x_i \geq \tau/2$ for every $i \in \sigma$. Define $C'(x) \in \{-1, 1\}^n$ to be one of the preimages of $t(x)$ under f, that is,

$$(f_1(C'(x)), \ldots, f_k(C'(x))) = t(x).$$

Let W_p denote a $\{-1, 1\}$-valued random variable with mean p. The decoding algorithms $\mathcal{A}'_i(y)$ run $\mathcal{A}_i(y)$ internally and give their output as follows:

$$\mathcal{A}'_i(y) = \begin{cases} \text{Output } W_{(1-s_i)/2} & \text{if } \mathcal{A}_i(y) \text{ returns } 1 \\ \text{Output } -W_{(1+s_i)/2} & \text{if } \mathcal{A}_i(y) \text{ returns } -1 \end{cases}$$

Therefore, for every $x \in \{-1, 1\}^\sigma$ and for every $i \in \sigma$,

$$\begin{aligned} x_i \mathbb{E}[\mathcal{A}'_i(C'(x))] &= x_i \mathbb{E}\left[\frac{(1 + \mathcal{A}_i(C'(x)))}{2} W_{(1-s_i)/2} - \frac{(1 - \mathcal{A}_i(C'(x)))}{2} W_{(1+s_i)/2}\right] \\ &= \frac{x_i}{2} \mathbb{E}\left[\mathcal{A}_i(C'(x)) - s_i\right] \\ &= \frac{x_i}{2}(f_i(C'(x)) - s_i) \\ &= \frac{x_i}{2}(t(x)_i - s_i) \\ &\geq \frac{\tau}{4} \gtrsim \eta. \end{aligned}$$

Since the probability that $\mathcal{A}'_i(C'(x))$ queries any particular location of $C'(x)$ is still at most c/n, it follows that C' is a $(q, c, \Omega(\eta))$-smooth code. By Theorem 8, C' is also a $(q, \Omega(\eta/c), \Omega(\eta))$-LDC. ◀

5 From LDCs to outlaws

In this section we prove Theorem 5, the converse of our main result.

Proof of Theorem 5. By Theorem 8, $C : \{-1, 1\}^k \to \{-1, 1\}^n$ is also a $(q, q/\delta, \eta)$-smooth code. For each $i \in [k]$, let $\mathcal{B}_i$ be its decoder for the ith index. Let $\nu_i : 2^{[n]} \to [0, 1]$ be the probability distribution used by $\mathcal{B}_i$ to sample a set $S \subseteq [n]$ of at most q coordinates and let $f_{i,S} : \{-1, 1\}^n \to [-1, 1]$ be function whose value at $y \in \{-1, 1\}^n$ is the expectation of the random sign returned by $\mathcal{B}_i(y)$ conditioned on the event that it samples S. Since this value depends only on the coordinates in S, the function $f_{i,S}$ is a q-junta.

Fix an $i \in [k]$ and let $f_i : \{-1, 1\}^n \to [-1, 1]$ be the function given by $f_i = \mathbb{E}_{S \sim \nu_i}[f_{i,S}]$. Then, since a q-junta has degree at most q, so does f_i. We claim that f_i is $\delta/(q2^{q/2})$-smooth. Since the functions $f_{i,S} : \{-1, 1\}^n \to \{-1, 1\}$ are q-juntas, it follows from Parseval's identity that they have spectral norm at most $2^{q/2}$. Moreover, for each $j \in [n]$, we have $\Pr_{S \sim \nu_i}[j \in S] \leq q/(\delta n)$. Hence, since $f_{i,S}$ depends only on the coordinates in S, we have

$$\|D_j f_i\|_{\mathrm{sp}} \leq \sum_{S \ni j} \nu_i(S) \|f_{i,S}\|_{\mathrm{sp}} \leq \frac{q2^{q/2}}{\delta n},$$

which gives the claim. By (3), it holds for every $x \in \{-1,1\}^k$ and every $i \in [k]$ that

$$x_i f_i\big(C'(x)\big) \geq 2\eta. \tag{7}$$

Define the distribution μ to correspond to the process of sampling $i \in [k]$ uniformly at random and returning f_i. Let $\bar{g} = (f_1 + \cdots + f_k)/k$ be the mean of μ. We show that $\kappa_\mu(\eta) \geq \eta k$. To this end, let $l = \eta k$, let $\sigma : [l] \to [k]$ be an arbitrary map and define the functions $g_1, \ldots, g_l$ by $g_i = f_{\sigma(i)}$. Let $x \in \{-1,1\}^k$ be such that for each $i \in [l]$, we have $x_{\sigma(i)} = 1$ and $x_j = -1$ elsewhere. It follows from (7) that $f_{\sigma(i)}\big(C(x)\big) \in [2\eta, 1]$ for every $i \in [l]$ and that $f_i\big(C(x)\big) \leq 0$ for every other $i \in [k]$. Hence,

$$\begin{aligned}\Big\|\frac{1}{l}\sum_{i=1}^{l}(g_i - \bar{g})\Big\|_{L_\infty} &\geq \Big(\frac{1}{l}\sum_{i=1}^{l}(g_i - \bar{g})\Big)\big(C(x)\big) \\ &= \frac{1}{l}\sum_{i=1}^{l} f_{\sigma(i)}\big(C(x)\big) - \frac{1}{k}\sum_{i=1}^{k} f_i\big(C(x)\big) \\ &\geq 2\eta - \frac{l}{k} = \eta.\end{aligned}$$

If σ maps each element in $[l]$ to a uniformly random element in $[k]$, then $g_1, \ldots, g_l$ are independent, μ-distributed and satisfy

$$\mathbb{E}\left[\Big\|\frac{1}{l}\sum_{i=1}^{l}(g_i - \bar{g})\Big\|_{L_\infty}\right] \geq \eta,$$

which shows that $\kappa_\mu(\eta) \geq l$. Finally we can scale all the functions in μ to make them 1-smooth, and get a distribution $\tilde{\mu}$ over 1-smooth functions with $\kappa_{\tilde{\mu}}(\eta\delta/(q2^{q/2})) \geq \eta k$. ◀

6 Candidate outlaws

In this section we elaborate on the candidate outlaws mentioned in the introduction.

6.1 Incidence geometry

We begin by describing a variant of Corollary 6 based on a slightly different assumption and show conditions under which this assumption holds. Let p be an odd prime, let $\mathbb{F}_p$ be a finite field with p elements and let n be a positive integer. For $x, y \in \mathbb{F}_p^n$, the *line* with origin x in direction y, denoted $\ell_{x,y}$, is the sequence $(x + \lambda y)_{\lambda \in \mathbb{F}_p}$. A line is nontrivial if $y \neq 0$.

▶ **Corollary 16.** *For every odd prime p and $\varepsilon \in (0,1]$, there exist a positive integer $n_1(p,\varepsilon)$ and a $c = c(p,\varepsilon) \in (0, 1/2]$ such that the following holds. Let $n \geq n_1(p,\varepsilon)$ and k be positive integers. Assume that for every set $A \subseteq \mathbb{F}_p^n$ of size $|A| \leq k$, there exists a set $B \subseteq \mathbb{F}_p^n$ of size εp^n such that every nontrivial line through A contains at most $p-2$ points of B. Then, there exists a $(p-1, c, c)$-LDC sending $\{0,1\}^l$ to $\{0,1\}^{p^n}$, where $l = \Omega(c^2 k/\log(1/c))$.*

The proof uses the following version of Szemerédi's Theorem [28, Theorem 1.5.4] and its standard "Varnavides-type" corollary (see for example [30, Exercise 10.1.9]).

▶ **Theorem 17** (Szemerédi's theorem)**.** *For every odd prime p and any $\varepsilon \in (0,1]$, there exists a positive integer $n_0(p,\varepsilon)$ such that the following holds. Let $n \geq n_0(p,\varepsilon)$ and let $S \subseteq \mathbb{F}_p^n$ be a set of size $|S| \geq \varepsilon p^n$. Then, S contains a nontrivial line.*

▶ **Corollary 18.** *For every odd prime p and any $\varepsilon \in (0,1]$, there exists a positive integer $n_1(p,\varepsilon)$ and a $c(p,\varepsilon) \in (0,1]$ such that the following holds. Let $n \geq n_1(p,\varepsilon)$ and let $S \subseteq \mathbb{F}_p^n$ be a set of size $|S| \geq \varepsilon p^n$. Then, S contains at least $c(p,\varepsilon)p^{2n}$ nontrivial lines, that is,*

$$\Pr_{x\in\mathbb{F}_p^n, y\in\mathbb{F}_p^n\setminus\{0\}}\left[\{(x+\lambda y)_{\lambda=0}^{p-1}\} \subset S\right] \geq c(p,\epsilon).$$

Proof of Corollary 16. With some abuse of notation, we identify functions $f : \mathbb{F}_p^n \to \{-1,1\}$ with vectors in $\{-1,1\}^{\mathbb{F}_p^n}$. Let $\phi : \{-1,1\} \to \{0,1\}$ be the invertible map $\phi(\alpha) = (\alpha+1)/2$. For a function $f : \mathbb{F}_p^n \to \{-1,1\}$, let $\phi(f) : \mathbb{F}_p^n \to \{0,1\}$ be the function $\phi(f)(x) = \phi(f(x))$ and for $f : \mathbb{F}_p^n \to \{0,1\}$, define $\phi^{-1}(f) : \mathbb{F}_p^n \to \{-1,1\}$ analogously.

For every $x \in \mathbb{F}_p^n$, let $F_x : \{-1,1\}^{\mathbb{F}_p^n} \to \mathbb{R}$ be the degree-$(p-1)$ function

$$F_x(f) = \mathbb{E}_{y\in\mathbb{F}_p^n\setminus\{0\}}\left[\prod_{\lambda\in\mathbb{F}_p^*}\phi(f)(x+\lambda y)\right]. \tag{8}$$

Then, for a set $B \subseteq \mathbb{F}_p^n$, the value $F_x(\phi^{-1}(1_B))$ equals the fraction of all nontrivial lines $\ell_{x,y}$ through x of which B contains the $p-1$ points $\{x+\lambda y : \lambda \in \mathbb{F}_p^*\}$. If B has size at least εp^n, it thus follows from Corollary 18 that $\mathbb{E}_{x\in\mathbb{F}_p^n}[F_x(\phi^{-1}(1_B))] \geq c(p,\varepsilon)$. Moreover, since the monomials in the expectation of (8) involve disjoint sets of variables and can be expanded as

$$\prod_{\lambda\in\mathbb{F}_p^*}\phi(f)(x+\lambda y) = \frac{1}{2^q}\sum_{S\subseteq\mathbb{F}_p^*}\prod_{\lambda\in S} f(x+\lambda y),$$

it follows that each F_x is $2(1-p^{-n})$-smooth.

Let μ be the uniform probability distribution over all F_x. We claim that $\kappa_\mu(c(p,\varepsilon)) \geq k$, which implies the result by Theorem 4 since μ is supported by degree $(p-1)$-functions. For every set $A \subseteq \mathbb{F}_p^n$ of size $|A| \leq k$, let $B \subseteq \mathbb{F}_p^n$ be an arbitrary set as in the assumption of the corollary and let $f_A = \phi^{-1}(1_B)$. Let z be a uniformly distributed random variable over $\mathbb{F}_p^n$, let $z_1,\ldots,z_k$ be independent copies of z and let $A = \{z_1,\ldots,z_k\}$. Then, $F_{z_1},\ldots,F_{z_k}$ are independent μ-distributed random functions and since every nontrivial line through A meets B in at most $p-2$ points, we have $F_{z_i}(f_A) = 0$ for every $i \in [k]$. Hence,

$$\mathbb{E}\Big[\Big\|\frac{1}{k}\sum_{i=1}^k\big(F_{z_i}-\mathbb{E}[F_z]\big)\Big\|_{L_\infty}\Big] \geq \mathbb{E}\Big[\Big|\frac{1}{k}\Big(\sum_{i=1}^k\big(F_{z_i}-\mathbb{E}[F_z]\big)\Big)(f_A)\Big|\Big] \geq c(p,\varepsilon),$$

which gives the claim. ◀

The proof of the formal version of Corollary 6 (given below) is similar to that of Corollary 16, so we omit it. In the following, $\mathbb{PF}_p^{n-1}$ is the projective space of dimension $n-1$, which is the space of directions in $\mathbb{F}_p^n$. The formal version of Corollary 6 is then as follows.

▶ **Corollary 19.** *For every odd prime p and $\varepsilon \in (0,1]$, there exist a positive integer $n_1(p,\varepsilon)$ and a $c = c(p,\varepsilon) \in (0,1/2]$ such that the following holds. Let $n \geq n_1(p,\varepsilon)$ and k be positive integers. Suppose that for every set of directions $D \subset \mathbb{PF}_p^{n-1}$ of size $|D| \leq k$, there exists a set $B \subset \mathbb{F}_p^n$ of size $|B| \geq \varepsilon p^n$ which does not contain any lines with direction in D. Then, there exists a (p,c,c)-LDC sending $\{0,1\}^l$ to $\{0,1\}^{p^n}$, where $l = \Omega(c^2k/\log(1/c))$.*

Feasible parameters for Corollary 16

Proving lower bounds on k for which the assumption of Corollary 16 holds true thus allows one to infer the existence of $(p-1)$-query LDCs with rate $\Omega(k/N)$ for $N = p^n$, provided p and ε are constant with respect to n. We establish the following bounds, which imply the (well-known) existence of $(p-1)$-query LDCs with message length $k = \Omega((\log N)^{p-2})$.

▶ **Theorem 20.** *For every odd prime p there exists an $\varepsilon(p) \in (0,1]$ such that the following holds. For every set $A \subseteq \mathbb{F}_p^n$ of size $|A| \leq \binom{n+p-3}{p-2} - 1$, there exists a set $B \subseteq \mathbb{F}_p^n$ of size $\varepsilon(p)p^n$ such that every line through A contains at most $p-2$ points of B.*

The proof uses some basic properties of polynomials over finite fields. For an n-variate polynomial $f \in \mathbb{F}_p[x_1, \ldots, x_n]$ denote $Z(f) = \{x \in \mathbb{F}_p^n : f(x) = 0\}$. The starting point of the proof is the following standard result (see for example [29]), showing that small sets can be 'captured' by zero-sets of nonzero, homogeneous polynomials of low degree.

▶ **Lemma 21** (Homogeneous Interpolation)**.** *For every $A \subseteq \mathbb{F}_p^n$ of size $|A| \leq \binom{n+d-1}{d} - 1$, there exists a nonzero homogeneous polynomial $f \in \mathbb{F}_p[x_1, \ldots, x_n]$ of degree d such that $A \subseteq Z(f)$.*

The next two lemmas show that if f is nonzero, homogeneous and degree d, and if $a \in \mathbb{F}_p^*$ is such that $f^{-1}(a)$ is nonempty, then lines through $Z(f)$ meet $f^{-1}(a)$ in at most d points.

▶ **Lemma 22.** *Let $f \in \mathbb{F}_p[x_1, \ldots, x_n]$ be a nonzero homogeneous polynomial of degree d. Let $a \in \mathbb{F}_p^*$ be such that the set $f^{-1}(a)$ is nonempty. Then, every line that meets $f^{-1}(a)$ in $d+1$ points must have direction in $Z(f)$.*

Proof. The univariate polynomial $g(\lambda) = f(x + \lambda y)$ formed by the restriction of f to a line $\ell_{x,y}$ has degree at most d. By the Factor Theorem, such a polynomial must be the constant polynomial $g(\lambda) = a$ to assume the value a for $d+1$ values of λ. Since f is homogeneous, the coefficient of λ^d, which must be zero, equals $f(y)$, giving the result. ◀

The following lemma is essentially contained in [8].

▶ **Lemma 23** (Briët–Rao)**.** *Let $f \in \mathbb{F}_p[x_1, \ldots, x_n]$ be a nonzero homogeneous polynomial of degree d. Let $a \in \mathbb{F}_p^*$ be such that $f^{-1}(a)$ is nonempty. Then, there exists no line that intersects $Z(f)$, meets $f^{-1}(a)$ in at least d points and has direction in $Z(f)$.*

Proof. For a contradiction, suppose there exists a line $\ell_{x,y}$ through $Z(f)$ that meets $f^{-1}(a)$ in d points and has direction $y \in Z(f)$. Observe that for every $\lambda \in \mathbb{F}_p$, the shifted line $\ell_{x+\lambda y,y}$ also meets $f^{-1}(a)$ in d points. Hence, without loss of generality we may assume that the line starts in $Z(f)$, that is $x \in Z(f)$. Let $g(\lambda) = a_0 + a_1\lambda + \cdots + a_d\lambda^d = f(x + \lambda y) \in \mathbb{F}_p[\lambda]$ be the restriction of f to $\ell_{x,y}$. It follows that $a_0 = g(0) = f(x) = 0$ and, since f is homogeneous, that $a_d = f(y) = 0$. Moreover, there exist distinct elements $\lambda_1, \ldots, \lambda_d \in \mathbb{F}_p^*$ such that $g(\lambda_i) = f(x + \lambda_i y) = a$ for every $i \in [d]$. Then $g(\lambda) - a$ is a degree $d-1$ polynomial with d distinct roots. But it cannot be the zero polynomial since it takes value $-a$ when $\lambda = 0$. ◀

The final ingredient for the proof of Theorem 20 is the DeMillo–Lipton–Schwartz–Zippel Lemma as it appears in [10].

▶ **Lemma 24** (DeMillo–Lipton–Schwartz–Zippel)**.** *Let $f \in \mathbb{F}_p[x_1, \ldots, x_n]$ be a nonzero polynomial of degree d and denote $r = |\mathbb{F}_p|$. Then,*

$$|Z(f)| \leq \Big(1 - \frac{1}{r^{d/(r-1)}}\Big) r^n.$$

Proof of Theorem 20. Let $A \subseteq \mathbb{F}_p^n$ be a set of size $|A| \leq \binom{n+p-3}{p-2} - 1$. Let $f \in \mathbb{F}_p[x_1, \ldots, x_n]$ be a nonzero degree-$(p-2)$ homogeneous polynomial such that $A \subseteq Z(f)$, as promised to exist by Lemma 21. By Lemma 24, there exists an $a \in \mathbb{F}_p^*$ such that the set $B = f^{-1}(a)$ has size at least $|B| \geq p^n / p^{(2p-3)/(p-1)}$. By Lemma 22, every line that meets B in $p-1$ points must have direction in $Z(f)$, but by Lemma 23 no such line can pass through $Z(f)$. Hence, every line through A meets B in at most $p-2$ points. ◀

6.2 Uniformity of random Cayley hypergraphs

A second candidate for constructing outlaws comes from quasirandom properties of Cayley graphs and hypergraphs.

Random Cayley graphs and 2-query LDCs

For graphs, an important quasirandom property is spectral expansion. If for a regular graph G, we let $1 = \lambda_1(G) \geq \lambda_2(G) \geq \cdots \geq \lambda_n(G) \geq -1$ denote the eigenvalues of the normalized adjacency matrix, then the second eigenvalue is defined as $\lambda(G) = \max_{i\in\{2,\ldots,n\}} |\lambda_i(G)|$. The importance of this parameter stems from the fact that if it is small, then every large subset of vertices is connected to its complement by a large number of edges [27, 1], a property that sparse random graphs have with high probability (we refer to [17] for a survey on expander graphs).

A famous result of Alon and Roichman [2] asserts that random Cayley graphs require relatively low degree to be spectral expanders with good probability. For a finite group Γ and an element $g \in \Gamma$, the Cayley graph $\mathrm{Cay}(\Gamma, \{g\})$ is the 2-regular graph with vertex set Γ and edge set $\{\{u, gu\} : u \in \Gamma\}$, where in case $g^2 = 1$, all edges are doubled (parallel edges are thus allowed). For a multiset $S = \{g_1, \ldots, g_k\} \subseteq \Gamma$, the Cayley graph $\mathrm{Cay}(\Gamma, S)$ is the $2k$-regular graph formed by the disjoint union of the graphs $\mathrm{Cay}(\Gamma, \{g_1\}), \ldots, \mathrm{Cay}(\Gamma, \{g_k\})$.

▶ **Theorem 25** (Alon–Roichman Theorem). *For any $\varepsilon \in (0, 1)$ there exists a $c(\varepsilon) \in (0, \infty)$ such that the following holds. Let Γ be a finite group of cardinality n. Let $k \geq c(\varepsilon) \log n$ be an integer and let $g_1, \ldots, g_k$ be independent uniformly distributed elements from Γ. Then, with probability at least $1/2$, the Cayley graph $G = \mathrm{Cay}(\Gamma, \{g_1, \ldots, g_k\})$ satisfies $\lambda(G) \leq \varepsilon$.*

The link with Theorem 4 follows from the fact that the above result can equivalently be phrased as saying that the normalized adjacency matrix A_G of the random graph G as in Theorem 25 *concentrates* around its expectation. The normalized adjacency matrix of $\mathrm{Cay}(\Gamma, \{g_i\})$, denoted A_i, has expectation J/n, where J is the all-ones matrix, and A_G is the average $(A_1 + \cdots + A_k)/k$. The Schatten-∞ norm (also known as the spectral norm or operator norm) of a matrix B is given by $\|B\|_{S_\infty} = \sup\{x^{\mathsf{T}} B y : \|x\|_{\ell_2} \leq 1, \|y\|_{\ell_2} \leq 1\}$. Due to the characterization $\lambda(G) = \|A_G - J/n\|_{S_\infty}$, Theorem 25 says that the value

$$\Big\| \frac{1}{k} \sum_{i=1}^{k} (A_i - J/n) \Big\|_{S_\infty}$$

is small with good probability provided k is large enough. Good concentration is thus good for expansion. Our Theorem 4 implies that *poor* concentration is good for LDCs!

▶ **Corollary 26.** *Let Γ be a finite group of cardinality n and let $\varepsilon \in (0, 1)$. Let k be the largest positive integer such that with probability at least $1/2$, for independent uniformly distributed elements $g_1, \ldots, g_k \in \Gamma$, the graph $G = \mathrm{Cay}(\Gamma, \{g_1, \ldots, g_k\})$ satisfies $\lambda(G) > \varepsilon$. Then, there exists a $(2, 1, \Omega(\varepsilon))$-LDC sending $\{0,1\}^l$ to $\{0,1\}^n$, where $l = \Omega(\varepsilon^2 k / \log(1/\varepsilon))$.*

It is not hard to show that if Γ is Abelian, then any Cayley graph $G = \mathrm{Cay}(\Gamma, S)$ generated by $|S| \leq (\log|\Gamma|)/3$ group elements satisfies $\lambda(G) \geq 1/2$ [17, Proposition 11.5]. Together with Corollary 26, this fact implies the existence of 2-query LDCs of exponential length; arguably the most round-about way to prove this!

Below, we prove a more general version of Corollary 26, giving the existence of q-query LDCs from lower bounds on the required degree for uniformity of random q-uniform Cayley hypergraphs. For this, we gather the following definitions.

Regular hypergraphs and uniformity

A q-uniform hypergraph $H = (V, E)$ with vertex set V has as edge set E a family of unordered q-element multisets with possible parallel edges. For $u_1, \dots, u_q \in V$ let $e_H(u_1, \dots, u_q)$ denote the number of edges equal to $\{u_1, \dots, u_q\}$.[4] The *adjacency form* of H is the q-linear form $\overline{A}_H : \mathbb{R}^V \times \cdots \times \mathbb{R}^V \to \mathbb{R}$ given by $\overline{A}_H(1_{\{u_1\}}, \dots, 1_{\{u_q\}}) = e_H(u_1, \dots, u_q)$. The degree of a vertex $v \in V$ is defined by $\overline{A}_H(1_{\{v\}}, 1_V, \dots, 1_V)$ and H is k-regular if every vertex has degree exactly k, in which case its normalized adjacency form is $A_H = \overline{A}_H/k$. Observe that we obtain the usual definition of a regular graph for the case $q = 2$.

Given a regular hypergraph $H = (V, E)$ and a regular sub-hypergraph $J = (V, E')$ of H based on a multiset $E' \subseteq E$, we define the *uniformity of J relative to H* by

$$\Delta_H(J) = \max_{T_1,\dots,T_q \subseteq V} \frac{1}{|V|} \Big| A_J(1_{T_1}, \dots, 1_{T_q}) - A_H(1_{T_1}, \dots, 1_{T_q}) \Big|. \tag{9}$$

Observe that we get the usual notion of uniformity for graphs if H is the complete graph with all self-loops. We say that J *is ε-uniform relative to H* if $\Delta_H(J) \le \varepsilon$.

Cayley hypergraphs

A Cayley hypergraph over a finite group Γ is a disjoint union of hypergraphs with edge sets of the form $\{\pi_1(v), \dots, \pi_q(v)\}$, $v \in \Gamma$, where $\pi_1, \dots, \pi_q$ are permutations on Γ, and where the edge multiplicities are set such that

$$e_H(u_1, \dots, u_q) = \sum_{\sigma \in S_q} \sum_{v \in \Gamma} 1_{\{u_1\}}\big(\pi_{\sigma(1)}(v)\big) \cdots 1_{\{u_q\}}\big(\pi_{\sigma(q)}(v)\big). \tag{10}$$

This choice of multiplicities ensures that we always get a $(q!)$-regular hypergraph regardless of the choice of permutations. The permutations defining Cayley hypergraphs are given in terms of an integer vector $r \in (\mathbb{Z} \setminus \{0\})^q$ such that Γ has no elements of order r_j for every $j \in [q]$, so that for every $g \in \Gamma$ and $j \in [q]$, the map $u \mapsto u^{r_j} g$ is a permutation. For a q-tuple $\mathbf{g} = (g[1], \dots, g[q]) \in \Gamma^q$, we then define $\mathrm{Cay}^{(q)}(\Gamma, r, \mathbf{g})$ to be the hypergraph as above with $\pi_j(u) = u^{r_j} g[j]$. For a multiset $S = \{\mathbf{g}_1, \dots, \mathbf{g}_k\} \subseteq \Gamma^q$, we let $\mathrm{Cay}^{(q)}(\Gamma, r, S)$ be the $(q!)k$-regular hypergraph given by the disjoint union of $\mathrm{Cay}^{(q)}(\Gamma, r, \{\mathbf{g}_i\})$ for $i \in [k]$.

Examples

Notice that $\mathrm{Cay}^{(2)}(\Gamma, (1,1), (g,h))$ is the 2-regular Cayley graph generated by the element $g^{-1}h$. If $\Gamma = \mathbb{F}_p^n$ for some odd prime p, if $\mathbf{1} \in \mathbb{Z}^p$ is the all-ones vector and if $S = \{(0, y, 2y, \dots, (p-1)y) : y \in \mathbb{F}_p^n\}$, then the edges of $H = \mathrm{Cay}^{(p)}(\Gamma, \mathbf{1}, S)$ are all the affine lines in $\mathbb{F}_p^n$. Moreover, if $D \subseteq \mathbb{F}_p^n$ and $S' = \{(0, y, 2y, \dots, (p-1)y) : y \in D\}$, then the edges $J = \mathrm{Cay}^{(p)}(\Gamma, \mathbf{1}, S')$ are all the affine lines whose direction lies in D. Finally, if $\Delta_H(J) \le \varepsilon$, then for every set $T \subseteq \mathbb{F}_p^n$, the fraction of lines in T with direction in D is within ε of the fraction of all lines in T. In [8] it is shown that if $n \ge p^2$, then there is an absolute constant $c(p) \in (0,1)$ depending on p only such that if H and J are as above, then $\Delta_H(J) < c(p)$ implies that $|D| \ge \Omega(n^{p-1})$.[5]

[4] Curly brackets delimit *multisets*: unordered lists that may contain repeated elements.

[5] Their proof, however, does not show that the assumption of Corollary 19 holds for $|D| \ge \Omega(n^{p-1})$, as it relies on the construction of a combinatorial rectangle $T_1 \times \cdots \times T_p$ consisting of different sets.

Random q-uniform Cayley hypergraphs and q-query LDCs

Theorem 4 implies the following link between LDCs and random Cayley hypergraphs.

▶ **Corollary 27.** *Let Γ be a finite group of cardinality n and let $H = \mathrm{Cay}^{(q)}(\Gamma, r, S)$ be a q-uniform Cayley hypergraph. Let k be the largest positive integer such that with probability at least 1/2, if $\mathbf{g}_1, \ldots, \mathbf{g}_k$ are independent uniformly distributed random elements over S, the random Cayley hypergraph $J = \mathrm{Cay}^{(q)}(\Gamma, r, \{\mathbf{g}_1, \ldots, \mathbf{g}_k\})$ satisfies $\Delta_H(J) \geq \varepsilon$. Then, there exists a $(q, 1, \Omega(\varepsilon))$-LDC sending $\{0,1\}^l$ to $\{0,1\}^n$, where $l = \Omega(\varepsilon^2 k / \log(1/\varepsilon))$.*

Proof. Associate with every q-linear form $A : \mathbb{R}^\Gamma \times \cdots \times \mathbb{R}^\Gamma \to \mathbb{R}$ a homogeneous $q|\Gamma|$-variate degree-q function $f : \{-1,1\}^{q|\Gamma|} \to \mathbb{R}$ in the obvious way. It follows from (10) and the normalization of the adjacency forms that for every fixed $g \in \Gamma$, the functions f_g associated with the adjacency form of the hypergraph $\mathrm{Cay}^{(q)}(\Gamma, r, \{g\})$ are 1-smooth.

Moreover, for any q-linear form A on $\mathbb{R}^n$, it holds that

$$\frac{\max\left\{|A(x[1], \ldots, x[q])| : x[1], \ldots, x[q] \in \{-1,1\}^n\right\}}{\max\left\{|A(1_{T_1}, \ldots, 1_{T_q})| : T_1, \ldots, T_q \subseteq [n]\right\}} \geq 1.$$

The form A_J is the average of k independent identically distributed adjacency forms $A_1, \ldots, A_k$ that have expectation A_H. Letting $f_1, \ldots, f_k$ and $\bar{f}$ be the functions associated with $A_1, \ldots, A_k$ and A_H, respectively, it follows that

$$\begin{aligned}\mathbb{E}\left[\left\|\frac{1}{k}\sum_{i-1}^{k}(f_i - \bar{f})\right\|_{L_\infty}\right] &= \mathbb{E}\left[\max\left\{|(A_J - A_H)(x[1], \ldots, x[q])| : x[1], \ldots, x[q] \in \{-1,1\}^\Gamma\right\}\right] \\ &\geq \mathbb{E}[\Delta_H(J)] \\ &\geq \frac{\varepsilon}{2},\end{aligned}$$

where the last line follows from the fact that uniformity is nonnegative. The result now follows from Theorem 4. ◀

Spectral expansion and uniformity

For regular *graphs*, the famous Expander Mixing Lemma [17] shows that $\Delta(G) \leq \lambda(G)$ holds in general. Whereas the reverse inequality does not hold in general [11], the reason why Corollary 26 could be stated in terms of the second eigenvalue is that for Cayley graphs, a near-reverse inequality does hold: for some absolute constant $c \in (0, \infty)$, we have $\Delta(G) \geq c\lambda(G)$ [21, 11]. A second eigenvalue for hypergraphs, analogous to uniformity for hypergraphs, was defined and studied in [8], where first steps to generalize the Alon–Roichman Theorem were taken. While the Expander Mixing Lemma easily generalizes, it is unknown whether the analogue of [21, 11] holds for Cayley hypergraphs.

References

1 Noga Alon and Vitali D. Milman. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory. Series B*, 38(1):73–88, 1985.

2 Noga Alon and Yuval Roichman. Random Cayley graphs and expanders. *Random Structures & Algorithms*, 5, 1994. URL: `http://citeseer.ist.psu.edu/alon97random.html`.

3 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

4 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

5 Manuel Blum and Sampath Kannan. Designing programs that check their work. *Journal of the ACM*, 42(1):269–291, 1995.

6 Jop Briët. On embeddings of ℓ_1^k from locally decodable codes. *arXiv preprint: arXiv:1611.06385*, 2016.

7 Jop Briët, Assaf Naor, and Oded Regev. Locally decodable codes and the failure of cotype for projective tensor products. *Electronic Research Announcements in Mathematical Sciences (ERA-MS)*, 19:120–130, 2012.

8 Jop Briët and Shravas Rao. Arithmetic expanders and deviation bounds for random tensors. *arXiv preprint arXiv:1610.03428*, 2016.

9 Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *Journal of the ACM*, 45(6):965–981, 1998.

10 Gil Cohen and Avishay Tal. Two structural results for low degree polynomials and applications. *arXiv preprint arXiv:1404.0654*, 2014.

11 David Conlon and Yufei Zhao. Quasirandom Cayley graphs. *Discrete Analysis*, 6, 2017. Available at arXiv:1603.03025 [math.CO].

12 Zeev Dvir and Sivakanth Gopi. 2-Server PIR with sub-polynomial communication. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 577–584. ACM, 2015.

13 Klim Efremenko. 3-query locally decodable codes of subexponential length. In *Proceedings of the fourty-first annual ACM symposium on Theory of computing (STOC 2009)*, pages 39–44, 2009.

14 Oded Goldreich, Howard Karloff, Leonard J Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. In *Computational Complexity, 2002. Proceedings. 17th IEEE Annual Conference on*, pages 143–151. IEEE, 2002.

15 Parikshit Gopalan, Cheng Huang, Huseyin Simitci, and Sergey Yekhanin. On the locality of codeword symbols. *IEEE Trans. Information Theory*, 58(11):6925–6934, 2012. doi:10.1109/TIT.2012.2208937.

16 Sivakanth Gopi, Swastik Kopparty, Rafael Oliveira, Noga Ron-Zewi, and Shubhangi Saraf. Locally testable and locally correctable codes approaching the gilbert-varshamov bound. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2017)*, pages 2073–2091. SIAM, 2017.

17 Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43:439–561, 2006.

18 Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of the 32nd annual ACM symposium on Theory of computing (STOC 2000)*, pages 80–86. ACM Press, 2000.

19 Tali Kaufman and Madhu Sudan. Sparse random linear codes are locally decodable and testable. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 590–600. IEEE, 2007.

20 Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *J. of Computer and System Sciences*, 69:395–420, 2004. Preliminary version appeared in STOC'03.

21 Yoshiharu Kohayakawa, Vojtěch Rödl, and Mathias Schacht. Discrepancy and eigenvalues of Cayley graphs. *preprint*, 2016. Available at arXiv:1602.02291 [math.CO].

22 Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. High-rate locally-correctable and locally-testable codes with sub-polynomial query complexity. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016,*

Cambridge, MA, USA, June 18-21, 2016, pages 202–215, 2016. doi:10.1145/2897518.2897523.

23 Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. *Journal of the ACM (JACM)*, 61(5):28, 2014.

24 Shachar Mendelson and Roman Vershynin. Entropy and the combinatorial dimension. *Invent. Math.*, 152(1):37–55, 2003.

25 Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948. doi:10.1002/j.1538-7305.1948.tb01338.x.

26 Michel Talagrand. *Upper and lower bounds for stochastic processes*, volume 60 of *Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge. A Series of Modern Surveys in Mathematics [Results in Mathematics and Related Areas. 3rd Series. A Series of Modern Surveys in Mathematics]*. Springer, Heidelberg, 2014. Modern methods and classical problems.

27 Michael R. Tanner. Explicit concentrators from generalized n-gons. *SIAM Journal on Algebraic Discrete Methods*, 5(3):287–293, 1984.

28 Terence Tao. *Higher order Fourier analysis*, volume 142. American Mathematical Society, 2012.

29 Terence Tao. Algebraic combinatorial geometry: the polynomial method in arithmetic combinatorics, incidence combinatorics, and number theory. *EMS Surv. Math. Sci.*, 1:1–46, 2014.

30 Terence Tao and Van Vu. *Additive Combinatorics*. Cambridge University Press, 2006.

31 David Woodruff. New lower bounds for general locally decodable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(006), 2007.

32 David Woodruff. A quadratic lower bound for three-query linear locally decodable codes over any field. *J. Comput. Sci. Technol.*, 27(4):678–686, 2012. doi:10.1007/s11390-012-1254-8.

33 Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. In *Proceedings of the 39th annual ACM symposium on Theory of computing (STOC 2007)*, pages 266–274, 2007.

34 Sergey Yekhanin. Locally decodable codes. *Foundations and Trends in Theoretical Computer Science*, 6(3):139–255, 2012.

Constant-Rate Interactive Coding Is Impossible, Even In Constant-Degree Networks

Ran Gelles[*1] **and Yael T. Kalai**[2]

1 **Faculty of Engineering, Bar-Ilan University, Ramat-Gan, Israel**
ran.gelles@biu.ac.il
2 **Microsoft Research, Boston, USA**
yael@microsoft.com

Abstract

Multiparty interactive coding allows a network of n parties to perform distributed computations when the communication channels suffer from noise. Previous results (Rajagopalan and Schulman, STOC '94) obtained a multiparty interactive coding protocol, resilient to random noise, with a blowup of $O(\log(\Delta+1))$ for networks whose topology has a maximal degree Δ. Vitally, the communication model in their work forces all the parties to send one message at every round of the protocol, even if they have nothing to send.

We re-examine the question of multiparty interactive coding, lifting the requirement that forces all the parties to communicate at each and every round. We use the recently developed information-theoretic machinery of Braverman et al. (STOC '16) to show that if the network's topology is a cycle, then there is a specific "cycle task" for which any coding scheme has a communication blowup of $\Omega(\log n)$. This is quite surprising since the cycle has a maximal degree of $\Delta = 2$, implying a coding with a *constant blowup* when all parties are forced to speak at all rounds.

We complement our lower bound with a matching coding scheme for the "cycle task" that has a communication blowup of $\Theta(\log n)$. This makes our lower bound for the cycle task tight.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity, E.4 Coding and Information Theory

Keywords and phrases Interactive Communication, Coding, Stochastic Noise, Communication Complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.21

1 Introduction

In multiparty interactive communication, n parties, connected via some arbitrary network $G = (V, E)$, try to compute some function f of their private inputs by communicating messages over the network. *Coding for interactive communication* asks for coding schemes that succeed to compute any such function even when the communication may be noisy. A fundamental question in this field is finding the maximal *rate* such coding schemes can achieve[1], that is, what is the minimal amount of redundancy coding schemes must add in order to successfully compute any function f despite the noise.

* Part of this work was done while the author was at Princeton University. Supported in part by NSF grant CCF-1149888.

[1] The rate of a coding scheme is the ratio between the communication of a protocol that performs over a noiseless network, to the communication of the coding scheme for the same task, over the noisy network.

© Ran Gelles and Yael T. Kalai;
licensed under Creative Commons License CC-BY

The work of Rajagopalan and Schulman [13] gave an initial answer to this question, assuming stochastic noise (e.g., when each bit is being flipped independently with some fixed probability $\varepsilon < 1/2$): Let Δ be the maximal degree in G, then any (noiseless) protocol χ can be simulated with high probability over the noisy network by a protocol χ' with communication complexity $\mathsf{CC}(\chi') = \mathsf{CC}(\chi) \cdot O(\log(\Delta + 1))$. That is, for constant-degree networks such as the line or the cycle, the rate, $\mathsf{CC}(\chi)/\mathsf{CC}(\chi')$, is a constant bounded away from 0 while for highly-connected graphs such as the star or the complete graph, the rate goes to zero when n tends to infinity, i.e., the rate is $\Theta(1/\log n)$. The work of Alon et al. [2] shows that coding schemes with constant (non-zero) rate also exist for the complete graph, and other highly-connected graphs, hinting that it may be possible to achieve a constant rate coding scheme for any network G. This hope was terminated by Braverman et al. [3], showing that a rate of $\Theta(\log\log n/\log n)$ is maximal for a specific task over the star network.

All the above works assume that the communication over the network is performed in rounds, where at every round *all the parties speak*, that is, $2|E|$ symbols are being communicated—one symbol over each channel of the network. A natural question to ask is: Why is such an assumption justifiable? One interpretation is that these previous works try to optimize the *round complexity*, as opposed to the communication complexity, and hence the assumption that all parties send a message to all other connected parties in each round.

In this work, our goal is to optimize the *communication complexity* (as opposed to round complexity), and we ask whether similar bounds on the rate follow if we don't force all parties to speak at every round.

Surprisingly, we show that the rate of coding schemes when G is a cycle (assuming channels with large alphabets) is at most $O(1/\log n)$. This corresponds to a lower bound of $\Omega(\log n)$ on the communication blowup. Informally, our main theorem is the following:

▶ **Theorem 1** (main, informal)**.** *Let G be the cycle graph with n parties. Then, for any constant $\varepsilon < 1/2$ there exists a task whose communication complexity over the noiseless G is d while any coding scheme over any noisy graph G' (with noise parameter ε) that succeeds with high probability has communication complexity $\Omega(d \log n)$.*

The above theorem is quite surprising in light of the result of Rajagopalan and Schulman [13]: the maximal degree in the cycle graph is $\Delta = 2$, therefore the coding scheme of [13] (in the "everybody speaks" model) has a rate of $\Theta(1)$, regardless of the size of the network! In hindsight, the reason for this discrepancy is simple: the fact that everybody speaks in the model of [13] implies an inherent blowup in the communication of $O(n)$, which allows the parties to overcome errors. Indeed, assume that the "relevant" information for computing the function f progresses along the cycle: first p_1 sends a message to p_2 (while all the other parties have nothing to send in the meantime), only then p_2 has a message to send to p_3 and so on. While the "relevant" information is limited to a single edge on the network at any round, the fact that all the parties *must* speak at every round multiplies the effective communication by n both for the noiseless and noisy protocols, hence, it cancels out in the rate. On the other hand, this superfluous redundancy gives the parties the opportunity to correct previous errors in rounds where they are supposed to be idling if we weren't to force all the parties to speak at every round, and charge the parties according to the communication that actually happened.

For our lower bound we don't restrict the topology G' of the noisy graph, and allow any party to communicate with any other party (since anyways we count the actual communication, allowing the coding scheme to utilize any underlying graph just makes our lower bound stronger). Our lower bound actually works when the noise erases symbols instead of corrupting symbols (again, making the result stronger). The only "restrictive" assumption we have on the coding scheme is a fixed speaking order, independent of the inputs and the noise; see the "Communication Model" subsection below for a discussion regarding this assumption.

1.1 The Cycle Task

The noiseless task we use for Theorem 1 is an analog of the "pointer jumping" task over a cycle (see formal description in Section 2.5). Every party begins with a 2^n-ary tree of depth d, where each edge is labeled by a single bit. Each party begins at the root of its own tree, and the goal is to travel down the tree until it reaches a leaf.

It is most convenient to describe this task via the protocol that solves it. The parties are activated in a cyclic order (first p_1, then p_2, etc.). When p_i is activated, it receives a message of the form $(b_1, \ldots, b_n)$ from p_{i-1}, corresponding to the labels of the edges traversed by the parties in the previous n rounds (padding with zeros as necessary in the first $n-1$ rounds). Upon receiving this message $\ell = (b_1, \ldots, b_n)$ from p_{i-1}, p_i moves down from its current node to its ℓ-th child. Denoting by b the label of the edge it just took, p_i communicates to p_{i+1} the string $(b_2, \ldots, b_n, b)$. This process continues until all parties reach a leaf at depth d in their input tree. The output is the path each party took along its tree.

In addition to the lower bound on the communication blowup, we show a coding scheme that successfully computes the cycle task over a noisy network with rate $\Theta(1/\log n)$, matching the rate of our lower bound for the cycle task (up to a constant).

▶ **Theorem 2** (upper bound, informal). *For any constant $\varepsilon < 1/2$, there exists a coding scheme that solves the cycle task of depth d over noisy channels with large alphabet and error parameter ε. The coding scheme obtains a rate of $\Theta(1/\log n)$ and a success probability of $1 - 2^{-\Omega(d \log n)}$.*

1.2 Communication Model

For our communication model, we assume that protocols have a *fixed* order of speaking. That is, we can assume that the protocol works in rounds so that the party that speaks at round i is determined in advance, independently of inputs and noise. This assumption is not without loss of generality, but we claim here that lifting this assumptions trivializes the model.

A completely unrestricted model would let the parties determine, at any round, whether they speak or not (cf. adaptive protocols for the two party case [1]; see also [8]). Such a model trivializes coding in the multiparty scenario, as parties can "encode" information via the path that the message is sent through: say p_1 wants to send a single bit to p_2. If the bit is 0, then p_1 sends the message directly. If it is 1, he can send the bit through p_n (who will relay it to p_2). Now, even if noise occurs[2], p_2 can figure out the bit in certainty by the identity of the sender.

Another model, which is not completely unrestricted but still trivializes coding in our scenario, is described in [11]. There, parties are allowed to decide whether to send a message or not (and to whom) according to the transcript so far. On its surface, this restriction avoids the "path encoding" described above, as parties are not allowed to change the delivery path according to their inputs. Nevertheless, such a model still enables error correction via "path selection", since the transcript still depends on the inputs. To give a simple example, assume a noiseless protocol in which the parties speak in order (p_1 sends a bit to p_2, then p_2 sends a bit to p_3, and so on). Such a protocol can be easily simulated over a noisy network in the [11] model: After p_i sends a bit to p_{i+1} the latter sends the bit back either directly (if it was a 0), or through p_{i-1} (if it was a 1); note that this decision is made as a function of the observed (possibly noisy) transcript, and thus it is allowed in that model. Now p_1 knows if its

[2] As long as we do not allow a stronger type of noise, i.e., insertions and deletions, see [4].

original bit reached p_{i+1} correctly or not and either retransmits the bit, or sends a message to p_{i+2} (who forwards it to p_{i+1}) to indicate that the bit was transferred correctly, and the simulation can move on to simulating the next bit of the noiseless protocol. In other words, this model reduces bit flips into erasures, and performing error correction from erasures with rate $1-\varepsilon$ is fairly simple if the model allows changing the order of the speaking according to the observed noise.

To conclude, we show that there is a strong relation between the order of speaking and the obtained coding rate. On one hand, allowing the order of speaking to change adaptively, allows trivial coding schemes. On the other hand, fixing the order of speaking allows us to show an $\Omega(\log n)$ lower bound on the blowup for the cycle task. It is however possible that worse rates are possible for other tasks. In fact, we conjecture that the blowup can get as high as $\Omega(n)$ in specific situations, as a function of the "mismatch" between the order of speaking in the noiseless protocol and the coding scheme.

▶ **Conjecture 3.** *There exists a topology G and a noiseless protocol χ with a fixed order of speaking for which any coding scheme χ' with fixed order of speaking has rate at most $O(1/n)$.*

Our findings are reminiscent of the two-party case: if the simulation has a fixed order, the order of speaking in the original scheme determines the maximal rate of the coding; specifically, it is conjectured that there exists a protocol whose simulation has rate bounded away from 1. On the other hand, if the simulation is allowed to be adaptive, better rates (that approach 1) can be achieved. See discussion in [12, 9].

1.3 On Binary vs. Large Alphabet

While our main result (Theorem 1) assumes that the parties communicate symbols from a large alphabet, we also obtain a lower-bound for the case where the parties communicate bits, i.e., use a binary alphabet. Typically, constructing coding schemes over the binary alphabet is harder than constructing such schemes over a large alphabet. However, our result is a lower-bound rather than a coding scheme, and it is not necessarily so that the binary-case is stronger (nor is more difficult to obtain).

Nevertheless, the setting of binary channels and the setting of large-alphabet channels seem *incomparable*, since the alphabet applies both to the original (noiseless) protocol and to the coded (noisy) protocol. We elaborate on this in Section 5.

We extend our lower bound result also to the case where the noiseless protocol and the coding scheme are binary. Specifically, we show a lower bound of $\tilde{\Omega}(\log n)$ on the blowup of the communication for binary coding scheme over the star network (where the $\tilde{\Omega}$ notation means neglecting $\log\log n$ terms). Informally, the theorem is the following.

▶ **Theorem 4** (binary case, informal)**.** *Let G be the star graph with n parties. Then, for any constant $\varepsilon < 1/2$ there exists a task whose communication complexity over the noiseless G is d while any coding scheme (with fixed order) over any noisy graph G' (with noise parameter ε) that succeeds with high probability has communication complexity $\tilde{\Omega}(d \log n)$.*

We stress that the above theorem is incomparable to the result of [3]: in our model parties may speak in an arbitrary (but fixed) order and are not forced to speak at every round. The task in consideration is the generalized jumping pointer described in [3]. The proof of Theorem 4 follows by combining the techniques developed in this paper for the cycle task with the techniques of [3] in quite a straightforward way, and we omit the details here.

1.4 Overview of our Techniques

The proof of our lower bound uses techniques from [3] for bounding the progress of a coding scheme χ' in simulating a noiseless protocol χ. As in [3], we use the notion of *cutoff* (Definition 11) that measures for any partial transcript of χ', how many cycles of the noiseless protocol χ are still not-simulated: when the cutoff is k, then the last $d-k$ cycles of χ are not simulated by the given transcript. More accurately (but still very informally) the transcript gives very little of information about the labels $\{b_i\}$ of the last $d-k$ cycles.

We show that any coding scheme that solves the cycle task with high probability must produce transcripts whose cutoff is $\approx d$, in expectation. Then, we show that for any segment in which the coding scheme communicates $O(n \log n)$ symbols, the cutoff advances by at most $O(1)$ cycles in expectation. Namely, let π be some fixed previous communication (including erasures), and let Π^{new} be the random variable describing the next $O(n \log n)$ symbols communicated by the coding scheme χ' (including erasures), then

$$\mathbb{E}[\mathsf{cutoff}(\pi \circ \Pi^{new}) \mid \mathsf{cutoff}(\pi) = k] \leq k + O(1).$$

In order for χ' to achieve an expected cutoff of $\approx d$, which is crucial for being correct with high probability, the coding scheme must communicate at least $\Omega(dn \log n)$ symbols, yielding a rate of $O(1/\log n)$.

The reason for the restricted progress in the cutoff is that many parties do not send any useful information in the segment Π^{new}, and that the next "move" (in the input tree) of each party depends on the moves of *all* the parties in the previous cycle. This means that most parties are missing a lot of crucial information in order to advance more than a constant number of levels in their input tree. Bounding the exact information sent by the parties (and thus the expected increase in the cutoff) is performed via the machinery of [3].

Showing that many parties give no information in any segment of $O(n \log n)$ rounds in our setting is a main technical difference from [3]. In the model of [3] all parties speak at every round, thus when the coding scheme communicates $O(n \log n)$ symbols we know that this communication is evenly spread—every party communicates exactly $O(\log n)$ symbols. In our setting, it is possible that the communication is evenly spread, but it is also possible that all $O(n \log n)$ symbols are communicated by a single party (or any other pattern in between). In the latter case, even if the noise targets the single party that speaks, that party could still convey $O(n \log n)$ bits of information by encoding its message via a standard error-correction code. Nevertheless, we show that there is a large set of parties that do not communicate any information in the new segment: either they don't speak at all, or they speak very little and their entire communication is completely erased by the noise. Furthermore, previous communication of these parties contains very little information on their labels in the last $d-k$ cycles to begin with.

The existence of this set of "erased" parties implies that the non-erased parties in this segment don't know how to proceed in their input tree, and their communication in that segment is "irrelevant" to the progress of the protocol, even if it is not erased by the noise. Indeed, assume a party's current node in its input tree is given, and assume that the party doesn't know which of its children it should go to next. The best that a party can do is to send all the labels below its current node. However, due to the fact that each node has 2^n children, that party cannot communicate more than $O(1)$ levels below its current node even if it gets to speak all the $O(n \log n)$ symbols in the next segment.

Naturally, the actual proof is more complex, since the party has some prior information about the children it should go (due to communication in previous rounds). This means that the children are not equiprobable and the party can communicate more information about

(the labels of) more probable children. Still, since the arity of the input tree is so large and since the information on the next children it should take is rather little, the party will be able to communicate information on the labels of only $O(1)$ levels below its current node (in expectation).

1.5 Other Related Work

The field of coding for interactive communication was initiated by Schulman [14, 15] who formalized the question for the two-party case and developed basic techniques used for solving this task, either when the noise is stochastic (where each bit is flipped with some constant probability) or adversarial (where any subset of up to ε-fraction of the bits can be flipped). Later works in the two-party setting improve on the computational efficiency, success probability, and achievable rate of coding schemes. We refer the reader to [6] for a survey on interactive coding.

As mentioned above, the interactive coding in the multiparty case was initiated by Rajagopalan and Schulman [13] for the random noise case. Efficiency for this setting is obtained by Gelles, Moitra and Sahai [7]. The works of Alon et al. [2] and of Braverman et al. [3] identify the maximal rate obtainable over the complete graph and the star (and provide efficient schemes that obtain such a rate). The case of adversarial noise is considered by Jain, Kalai and Lewko [11] providing a scheme for topologies that have a star as a subgraph, that withstands $O(1/n)$-fraction of adversarial noise and blows up the communication by only a constant. The work of Hoza and Schulman [10] provides a coding scheme for any topology $G = (V, E)$ that withstands $O(1/n)$-fraction of noise and obtains a rate of $\Theta(n/|E| \log n)$.

2 Preliminaries: Notations, Model, Coding Schemes

2.1 Notations and Basic Properties

For $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, 2, \ldots, n\}$. The log() function is taken to base 2.

▶ **Definition 5.** The Hamming distance $\Delta(\sigma, \sigma')$ of two strings $\sigma = \sigma_1 \ldots \sigma_m$ and $\sigma' = \sigma'_1 \ldots \sigma'_m$ of length m over an alphabet S, is the number of positions i such that $\sigma_i \neq \sigma'_i$.

Given any tree $\mathcal{T}$ of depth N, we denote its first k levels by $\mathcal{T}^{\leq k}$ and its $N - k$ last levels by $\mathcal{T}^{>k}$. Given a path $z = (e_1, e_2, \ldots)$, we denote by $\mathcal{T}[z]$ the subtree of $\mathcal{T}$ rooted at the end of the path that begins at the root of $\mathcal{T}$ and follows the edge-sequence z. The above notation composes for sets of trees, e.g., if $\vec{\mathcal{T}} = (\mathcal{T}_1, \mathcal{T}_2, ...)$ is an array of trees and $\vec{z} = (z_1, z_2, \ldots)$ is an array of paths, then we let $\vec{\mathcal{T}}^{\leq k}$ denote the array $(\mathcal{T}_1^{\leq k}, \mathcal{T}_2^{\leq k}, ...)$ and $\vec{\mathcal{T}}[\vec{z}]$ the array $(\mathcal{T}_1[z_1], \mathcal{T}_2[z_2], ...)$, etc.

As a rule, we use small letters to denote specific values (e.g., the input x_i given to party i), and capital letters to denote the corresponding random variables (i.e., X_i for the random variable describing the input of the i-th party, when the inputs are drawn from some given distribution).

2.2 Multiparty Interactive Communication and Protocols

We assume an undirected network $G = (V, E)$ of $n = |V|$ parties, $p_1, \ldots, p_n$, where p_i is connected to p_j if and only if $(i, j) \in E$. Each party is given an input x_i, and is assumed to output $f_i(x_1, \ldots, x_n)$ at the end of the protocol.

A *protocol* dictates to each party what is the next symbol to send and over which channel, given the party's input, the round number, and all the symbols that the party has received

so far. After a fixed and predetermined number of rounds, the protocol terminates and each party outputs a value as a function of its input and observed transcript. We assume that the order of speaking is *fixed* and is independent of the party's inputs and the noise. That is, it is determined in advance which channel is utilized at each round.

2.3 Noisy and Noiseless Networks

For the noiseless network, we focus on the *cycle network*. In the cycle, each party p_i is connected to p_{i-1} and p_{i+1} (all indices are modulo n).

For showing lower bounds over the noisy network we allow the parties to utilize the complete graph, avoiding any limitation on the protocol (since limiting the connectivity may harm the rate artificially). For our upper bound (coding scheme) the underlying topology is still the complete graph, however, the specific scheme we show does not need to communicate over all possible links—it communicates only over the cycle subgraph.

In a noisy network, each channel is assumed to suffer from random noise. For our lower bound we will assume each channel is a large-alphabet *erasure channel* EC_ε with erasure probability ε.

▶ **Definition 6.** For $\varepsilon \in [0,1]$ and a finite set Σ, the erasure channel over alphabet Σ is a random function $\mathsf{EC}_\varepsilon : \Sigma \to \Sigma \cup \{\perp\}$ which turns each input symbol into an erasure mark $\perp$ with probability ε, or otherwise keeps the symbol intact. When a channel is accessed multiple times, each instance is independent.

When considering upper bounds (coding schemes), channels with random noise are too weak (i.e., they can be reduced to erasure channels with high probability). Therefore, for our scheme we will assume a stronger type of noise we name semi-adversarial. Here, the transmissions that will be corrupted are determined in a random manner, however the received symbol of a corrupted transmission is determined *adversarially*; see discussion in Section 3.

▶ **Definition 7.** For $\varepsilon \in [0,1]$ and a finite set Σ, the semi-adversarial noisy channel over alphabet Σ is a random function $\mathsf{SAC}_\varepsilon : \Sigma \to \Sigma$ which corrupts any input symbol with probability ε, independently per instance. Once a symbol is corrupted, it may turn into any symbol in Σ, determined adversarially by the channel (possibly, all the corrupted symbols are chosen in a dependent manner).

2.4 Communication Complexity

For any protocol χ communicating symbols from an alphabet Σ, denote by $|\chi|$ the maximum number of symbols communicated by any execution of χ. Since we assume the order of speaking is fixed regardless of the inputs (and noise), each execution of χ has exactly $|\chi|$ number of symbols communicated. The communication complexity of χ is given by $\mathsf{CC}(\chi) = |\chi| \cdot \log|\Sigma|$.

2.5 The Cycle Task

In this section we define the cycle task and discuss a simple protocol that solves it over the noiseless cycle network.

Recall we have n parties $\{p_1, \ldots, p_n\}$ where each p_i receives the input x_i. We assume each input x_i is a labeled $|\Sigma|$-ary tree of depth d, where $\Sigma = \{0,1\}^n$ and each edge in the tree is labeled by a single bit.

The output of p_i is a simple root-to-leaf path (of length d) denoted by path_i, and the complete task output is denoted by $\mathsf{path} = (\mathsf{path}_1, \ldots, \mathsf{path}_n)$. We define the output in an inductive manner. For $i \in [n]$ and $j \in [d]$, let $\mathsf{path}_i(j) \in \{0,1\}^n$ denote the (index of the) j-th edge of path_i. Moreover, let $b_i(j) \in \{0,1\}$ denote the label of the edge that corresponds to $\mathsf{path}_i(j)$. For the induction basis, assume $b_i(j) = 0$ for all $i \in [n]$ and $j \leq 0$.

For $j \geq 1$, and for $i \in [n]$ we define $\mathsf{path}_i(j)$ as a function of $\{\mathsf{path}_{i'}(j')\}_{(j',i')<(j,i)}$, where $(x,y) < (u,v)$ holds if $x < u$ or if both $x = u$ and $y < v$; note that this implies a total order on pairs (j,i). The value of $\mathsf{path}_i(j)$ is given by the labels $b_{i'}(j')$ for the $n-1$ pairs (j',i') preceding (j,i) according to the total order we defined. Namely,

$$\mathsf{path}_i(j) = (b_{i+1}(j-1), b_{i+2}(j-1) \ldots, b_n(j-1), b_1(j), \ldots, b_{i-2}(j), b_{i-1}(j)).$$

Note that the cycle task can be solved by a simple protocol as described in Section 1. The protocol works in "cycles" where each such cycle means repeating the following process for $p_1, p_2, \ldots, p_n$ in order. During the j-th cycle p_i sends to p_{i+1} the value of $\mathsf{path}_i(j)$ along with the label $b_i(j)$ of the edge it just took. Now p_{i+1} can infer the value of $\mathsf{path}_{i+1}(j)$, and obtain the bit $b_{i+1}(j)$ labeling that edge in its input x_{i+1}. It follows that after d such "cycles" all parties reach a leaf at level d in their input, and can output path_i. Assuming the parties communicate symbols from Σ, the protocol communicates dn symbols[3] and has a communication complexity of dn^2 bits. It can be verified that the communication complexity of solving the cycle task is $\Theta(dn^2)$.

For our lower bound, we assume the inputs $X = (X_1, \ldots, X_n)$ are sampled so that each label is uniform in $\{0,1\}$. We are looking for coding schemes that solve the above task with high probability over the inputs X, the noise and the randomness of the coding scheme.

3 Upper Bound: A Coding Scheme For The Cycle Task With Blowup $\Theta(\log n)$

Before showing a lower bound of $\Omega(\log n)$ on the communication blowup of the cycle task over noisy networks, let us provide a sketch for a coding scheme that achieves a communication blowup of $\Theta(\log n)$, rendering our lower bound tight for the cycle task. The key idea is that repeating each symbol for $\Theta(\log n)$ times reduces the error probability to polynomially small in the number of parties. Then, the event of an error is so rare that standard coding techniques (that recover from small amount of errors) succeed with overwhelming probability.

When considering random noise over large alphabet, notice that the analog of the binary-symmetric-channel—a channel that uniformly picks the corrupted symbol—is too weak. Indeed, the parties could use only a small fraction of the symbol space in order to "catch" errors with high probability, thus essentially reducing the noise model into the case of erasures, while keeping the asymptotic rate the same up to a constant (see, for instance, the blueberry code technique in [5]).

Hence, our upper bound is defined in the somewhat stronger noise-model, which we call *semi-adversarial*, formally defined in Definition 7. In this noise model, each symbol is corrupted with probability ε, independently across different symbols. However, once a symbol is corrupted, the output symbol of the channel is chosen *adversarially*, in a worst case manner.

[3] In fact, it is enough to use $\Sigma = \{0,1\}^{n-1}$. We will neglect this issue as it doesn't change the asymptotic behaviour of the communication complexity, nor the asymptotic rate of related coding schemes.

3.1 Coding Scheme For The Cycle Task

The construction of our coding utilizes a primitive known as tree codes (see [15]; also see [6]).

▶ **Definition 8.** A β-ary tree code of depth γ, distance α and alphabet σ is a prefix code $TC : [\beta]^{\leq\gamma} \to \sigma^{\leq\gamma}$ that satisfies the following. For any two strings $x, y \in [\beta]^{\ell}$ of the same length $\ell \leq \gamma$ whose first difference is at the i-th coordinate,

$$\Delta(TC(x), TC(y)) \geq \alpha(\ell - i + 1),$$

where $\Delta(\cdot,\cdot)$ is the Hamming distance.

Schulman [15] showed that infinite-depth tree codes exist, and described the tradeoff between their distance and arity to their alphabet size.

▶ **Lemma 9** ([15]). *For any fixed $\beta \in \mathbb{N}$ and $\alpha \in (0,1)$, there exists a finite alphabet σ of size $|\sigma| = \beta^{O(1/(1-\alpha))}$ which suffices to construct a β-ary tree code with distance α and any depth.*

Our coding scheme, denoted by χ', uses tools from [13], and adapts them to our communication-model in which the parties are not forced to speak at every round. Let χ be the noiseless protocol for the cycle task described in Section 2.5. Our coding scheme χ' simulates χ step by step, sending each symbol that χ sends using two levels of coding (tree code and repetition code). While the decoding cannot guarantee that a party correctly decodes *all* the symbols sent to him so far, the symbols that were sent earlier in the protocol will be decoded correctly with an increasing probability. The party can then verify that the symbols he has already sent during previous rounds are consistent with his current understanding of the decoded incoming transmissions. In case they are not, the party will transmit a special $\mathcal{B}$ symbol whose meaning at the recipient is to "delete" the last (non-$\mathcal{B}$) symbol it has received. By sending multiple $\mathcal{B}$ symbols, the party can delete any incorrect suffix of his outgoing transmissions, until they become consistent with his (current view of his) incoming transmissions.

In the coding scheme χ' the parties communicate over channels with alphabet of size $(|\Sigma| + 1)$ that corresponds to all the symbols of χ and the additional "back" symbol $\mathcal{B}$. We assume a tree code with input alphabet $O(\Sigma)$ (specifically, a $(|\Sigma| + 1)$-ary tree), distance $\alpha > \varepsilon$, and output alphabet of size $|\Sigma'| = |\Sigma|^{O_\varepsilon(1)}$. Such a tree code exists due to Lemma 9. The coding scheme is described in Figure 1.

▶ Remark. In Figure 1, "sending a symbol" means sending $k = O_\varepsilon(\log n)$ repetitions of the same symbol using a repetition code with failure probability at most n^{-10}.

▶ **Theorem 10.** *For any $\epsilon < 1/2$, the coding scheme χ' has rate $\Theta_\varepsilon\left(\frac{1}{\log n}\right)$ and success probability $\geq 1 - 2^{-\Omega_\varepsilon(d \log n)}$, assuming the communication is over a SAC_ε network.*

4 The Lower Bound

In this section we give an outline of the proof of our lower bound. The complete description as well as detailed proofs are deferred to the full paper. Following [3], we define the notion of *cutoff* which measures the progress in simulating the cycle task. We show that the cutoff of a simulation is correlated with the length of the correct simulated output, in the sense that if the cutoff is k, it is improbable that the simulation gives an output whose correct prefix is of length more than k. Hence, if a simulation is correct with high probability, the implied cutoff must be high (i.e., around d).

The coding scheme χ'.

1. Repeat the following for $d' = 100d$ times.
2. For $i = 1$ to n, perform the following for p_i:
   - **a.** Let $y \in (\Sigma')^{\leq d'}$ be all the received communication from p_{i-1} in all the previous rounds.
     - **i.** Decode y via the tree code to obtain $x \in (\Sigma \cup \{\mathcal{B}\})^{\leq d'}$, i.e., set $x = TC^{-1}(y)$.
     - **ii.** Parse x to obtain $x' = \mathsf{Parse}(x)$.
       The function $\mathsf{Parse}(x)$ is defined in the following manner: Process x symbol-by-symbol in order. When processing a symbol from Σ, copy it to the output register. When processing a $\mathcal{B}$ symbol, delete the last non-deleted symbol in the output register. For instance, the string '$abd\mathcal{B}ccc\mathcal{B}d\mathcal{B}\mathcal{B}d$' is parsed to the string '$abcd$'.
   - **b.** Let $z \in (\Sigma \cup \{\mathcal{B}\})^{\leq d'}$ be all the symbols communicated by p_i so far during the protocol (before the tree-code encoding); let $z' = \mathsf{Parse}(z)$. p_i checks the consistency of its parsed incoming string x' and its parsed outgoing transmissions z'. The consistency is checked according to what p_i should have communicated over the noiseless χ, given the communication x'.
     If all its (parsed) outgoing messages z' are consistent with the (parsed) incoming messages, the next symbol to be sent, σ, is determined according to χ (if χ has already terminated, set $\sigma = 0$).
     If p_i finds an inconsistency, the next symbol to be sent is $\sigma = \mathcal{B}$.
   - **c.** p_i encodes the next symbol using the tree code, that is, it sends to p_{i+1} the last symbol of $TC(z \circ \sigma)$.

Figure 1 The coding scheme χ' for the Cycle Task.

Recall that x_i is the input of the i-th party, and X_i is the random variable describing it; similarly, π is used to describe a specific (observed) transcript while Π is the corresponding random variable. Also recall that the output of the i-th party is path_i describing the root-to-leaf path that the party traversed along x_i. Finally, recall that we denote by $\mathsf{path}_i(k)$ the first k edges in path_i and by $x_i[\mathsf{path}_i(k)]$ the subtree of x_i rooted at the end of $\mathsf{path}_i(k)$.

▶ **Definition 11** (Cutoff)**.** For any transcript π, and any input $x = (x_1, \ldots, x_n)$, the *cutoff of the protocol*, denoted by $\mathsf{cutoff}(\pi, x)$, is the minimal number k, such that

$$\sum_{i=1}^{n} I(X_i[\mathsf{path}_i(k)] \mid \Pi = \pi, \mathsf{PATH}(k) = \mathsf{path}(k)) \leq 0.01n. \tag{1}$$

We note that if $\mathsf{cutoff}(\pi, x) = k$ then for any x' such that $x'^{\leq k} = x^{\leq k}$, it holds that $\mathsf{cutoff}(\pi, x') = k$. Furthermore, the cutoff is only a function of the path up to level k, that is, if $\mathsf{cutoff}(\pi, x) = k$ then for any input x' that has the same $\mathsf{path}(k)$ it holds that $\mathsf{cutoff}(\pi, x') = k$; This property allows us to abuse notation and write $\mathsf{cutoff}(\pi, \mathsf{path}(k)) = k$, when the path is fixed but we do not care about the specific input.

The following proposition shows that in order for a protocol to output the correct value with high probability, the cutoff (given the complete transcript) must be $\approx d$. Hence, protocols that succeed with high probability must produce transcripts whose cutoff is large in expectation.

▶ **Proposition 12.** *Fix a protocol that solves the cycle task of depth d over a network with n parties (with large enough n), that succeeds with probability at least $1/5$ on average, i.e., a protocol for which $\Pr_{X,\Pi}[\textit{correct output}] \geq 1/5$. Then,*

$$\mathbb{E}_{X,\Pi}[\mathsf{cutoff}(\Pi, X)] \geq \frac{d}{10}.$$

Our main theorem shows that in order to obtain a coding with such a high cutoff (which is required for high success probability) a communication blowup of $\Omega(\log n)$ is necessary.

▶ **Theorem 13.** *For any $\varepsilon \in (0,1)$ there exists a constant $c = c(\varepsilon)$ such that for large enough n, any protocol that solves the cycle task of depth d over a network with n parties communicating less than $cd \cdot n \log n$ symbols assuming each communication channel is an EC_ε, has a success probability at most $1/5$.*

The main idea is to show that $O(n \log n)$ symbols sent by the simulation can increase the cutoff by at most $O(1)$, in expectation. That is, $O(\log n)$ cycles of the simulation are required in order to advance $O(1)$ cycles of the original protocol, giving a rate of $O(1/\log n)$.

Assume that given the (partial) observed transcript π and some path $\mathsf{path}(\ell)$, the cutoff of the coding scheme is ℓ, that is, $\mathsf{cutoff}(\pi, \mathsf{path}(\ell)) = \ell$. Then, assume we let the coding scheme communicate another $\delta \cdot n \log n$ symbols for some parameter $\delta = \delta(\varepsilon)$ we set later. We denote these new observed (potentially erased) symbols by Π^{new}; This is a random variable that depends on the noise and the randomness of the protocol. The claim is that the new cutoff (i.e., with respect to $\pi \circ \Pi^{new}$), is bounded by $\ell + O(1)$ in expectation.

▶ **Proposition 14.** *For any $\ell \leq d$, any $\mathsf{path}(\ell)$ and any transcript π,*

$$\mathbb{E}[\mathsf{cutoff}(\pi \circ \Pi^{new}, X) \mid \Pi = \pi, \mathsf{PATH}(\ell) = \mathsf{path}(\ell), \mathsf{cutoff}(\pi, \mathsf{path}(\ell)) = \ell] \leq \ell + 500.$$

With the above proposition, the proof of the main theorem is immediate.

Proof of Theorem 13. Assume χ is a coding scheme that succeeds with probability at least $1/5$. Proposition 12 claims that the expected cutoff at the end of the protocol χ is at least $d/10$.

On the other hand, assume toward contradiction that χ communicates less than $c \cdot d \cdot n \log n$ symbols. Split χ's transcript into segments of $\delta \cdot n \log n$ transmissions each. Using Proposition 14, the cutoff at the end of χ is bounded in expectation by

$$cd \cdot n \log n \cdot \frac{1}{\delta n \log n} \cdot 500 \leq \frac{500c}{\delta} d.$$

By choosing, say, $c < \delta/5000$, we get that the expected cutoff at the end of χ is strictly less than $d/10$, contradicting Proposition 12. ◀

The proof of Proposition 14 is rather involved and the details are deferred to the full version of this work.

5 Discussion: On the Rate vs. the Channel's Alphabet

In this section we discuss the effect of the channel's alphabet size on the obtainable rate. We can consider four independent settings: binary/large alphabet at the original (noiseless) scheme vs. binary/large alphabet at the coding scheme. For any $n \in \mathbb{N}$ and for $orig, code \in \{b, l\}$ let $c_{orig,code}(n)$ be the infimum over all possible n-party functions f of the maximal rate

Table 1 The relations between maximal rates of coding schemes with {binary, large}-alphabet, given the noiseless protocol uses {binary, large}-alphabet.

Noiseless Scheme χ	Coding Scheme χ': binary alphabet	Coding Scheme χ': large alphabet
binary alphabet	c_{bb}	$c_{bl} \geq \frac{c_{bb}}{\log \lvert\Sigma\rvert}$
large alphabet	$\Omega(c_{ll}) \leq c_{lb}$ $c_{bb} \leq c_{lb}$	$c_{ll} \geq c_{bl}$ $c_{ll} \geq \frac{c_{lb}}{\log \lvert\Sigma\rvert}$

obtainable when the original protocol χ for f is binary ($orig = b$) or with a large alphabet ($orig = l$) and the coding schemes χ' for f is binary or with a large alphabet ($code = b$ or $code = l$, respectively),

$$c_{orig,code}(n) = \inf_f \frac{\min_\chi \mathsf{CC}(\chi)}{\min_{\chi'} \mathsf{CC}(\chi')}.$$

The capacity of each setting—the maximal achievable rate in each setting—is defined to be the limit inferior of the above quantities when n tends to infinity,

$$c_{orig,code} = \liminf_{n\to\infty} c_{orig,code}(n).$$

We now explore relations between the four capacities. See Table 1 for a summary of the relations between the capacities of the different settings.

Any binary coding can be simulated by a large-alphabet coding by incurring a blowup of $\log|\Sigma|$, thus trivial relations are $c_{bl} \geq c_{bb}/\log|\Sigma|$ and $c_{ll} \geq c_{lb}/\log|\Sigma|$.

When the original protocol uses large alphabet, a large-alphabet coding can be reduced to a binary one by translating each symbol to a sequence of bits encoded with a standard error-correction code (so that the probability for the entire sequence to be decoded incorrectly is below ε; this can be done with a constant overhead). Thus $\Omega(c_{ll}) \leq c_{lb}$.

To see that $c_{lb} \geq c_{bb}$, note that we can convert the original large-alphabet protocol (that determines c_{lb}) into a binary one with the same communication complexity; this converted protocol may not be the hardest one for coding with a binary simulation, thus the rate we can achieve when coding it may be larger than the rate for the "worst" binary protocol, which determines c_{bb}. A similar reasoning yields $c_{ll} \geq c_{bl}$.

The above relations still allow c_{bb} to be either larger or smaller than c_{ll}, and their specific relation (as well as their feasibility with respect to a given underlying topology) remains an interesting open question.

References

1 Shweta Agrawal, Ran Gelles, and Amit Sahai. Adaptive protocols for interactive communication. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 595–599, 2016. doi:10.1109/ISIT.2016.7541368.

2 Noga Alon, Mark Braverman, Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Reliable communication over highly connected noisy networks. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, PODC '16, pages 165–173, 2016. doi:10.1145/2933057.2933085.

3 Mark Braverman, Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Constant-rate coding for multiparty interactive communication is impossible. In *Proceedings of the 48th*

Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, pages 999–1010, 2016. doi:10.1145/2897518.2897563.

4 Mark Braverman, Ran Gelles, Jieming Mao, and Rafail Ostrovsky. Coding for interactive communication correcting insertions and deletions. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 61:1–61:14, 2016. doi:10.4230/LIPIcs.ICALP.2016.61.

5 Matthew Franklin, Ran Gelles, Rafail Ostrovsky, and Leonard J. Schulman. Optimal coding for streaming authentication and interactive communication. *Information Theory, IEEE Transactions on*, 61(1):133–145, Jan 2015. doi:10.1109/TIT.2014.2367094.

6 Ran Gelles. Coding for interactive communication: A survey, 2015. URL: http://www.eng.biu.ac.il/~gellesr/survey.pdf.

7 Ran Gelles, Ankur Moitra, and Amit Sahai. Efficient coding for interactive communication. *Information Theory, IEEE Transactions on*, 60(3):1899–1913, March 2014. doi:10.1109/TIT.2013.2294186.

8 Mohsen Ghaffari, Bernhard Haeupler, and Madhu Sudan. Optimal error rates for interactive coding I: Adaptivity and other settings. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 794–803, 2014. doi:10.1145/2591796.2591872.

9 Bernhard Haeupler. Interactive Channel Capacity Revisited. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, FOCS '14, pages 226–235, 2014. doi:10.1109/FOCS.2014.32.

10 William M. Hoza and Leonard J. Schulman. The adversarial noise threshold for distributed protocols. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 240–258, 2016. doi:10.1137/1.9781611974331.ch18.

11 Abhishek Jain, Yael Tauman Kalai, and Allison Lewko. Interactive coding for multiparty protocols. In *Proceedings of the 6th Conference on Innovations in Theoretical Computer Science, ITCS '15*, pages 1–10, 2015. doi:10.1145/2688073.2688109.

12 Gillat Kol and Ran Raz. Interactive channel capacity. In *STOC '13: Proceedings of the 45th annual ACM Symposium on theory of computing*, pages 715–724, 2013. doi:10.1145/2488608.2488699.

13 Sridhar Rajagopalan and Leonard Schulman. A coding theorem for distributed computation. In *STOC '94: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 790–799, 1994. doi:10.1145/195058.195462.

14 Leonard J. Schulman. Communication on noisy channels: a coding theorem for computation. *Foundations of Computer Science, Annual IEEE Symposium on*, pages 724–733, 1992. doi:10.1109/SFCS.1992.267778.

15 Leonard J. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996. doi:10.1109/18.556671.

Parallel Repetition via Fortification: Analytic View and the Quantum Case

Mohammad Bavarian[*1], **Thomas Vidick**[†2], **and Henry Yuen**[‡3]

1 **Dept. of Mathematics, Massachusetts Institute of Technology, Cambridge, USA**
bavarian@mit.edu
2 **Dept. of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, USA**
vidick@cms.caltech.edu
3 **Dept. of Computer Science, University of California, Berkeley, USA**
hyuen@cs.berkeley.edu

Abstract

In a recent work, Moshkovitz [FOCS '14] presented a transformation on two-player games called "fortification", and gave an elementary proof of an (exponential decay) parallel repetition theorem for fortified two-player projection games. In this paper, we give an *analytic reformulation* of Moshkovitz's fortification framework, which was originally cast in combinatorial terms. This reformulation allows us to expand the scope of the fortification method to new settings.

First, we show *any* game (not just projection games) can be fortified, and give a simple proof of parallel repetition for general fortified games. Then, we prove parallel repetition and fortification theorems for games with players sharing quantum entanglement, as well as games with more than two players. This gives a new gap amplification method for general games in the quantum and multiplayer settings, which has recently received much interest.

An important component of our work is a variant of the fortification transformation, called "ordered fortification", that preserves the entangled value of a game. The original fortification of Moshkovitz does not in general preserve the entangled value of a game, and this was a barrier to extending the fortification framework to the quantum setting.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes

Keywords and phrases Parallel repetition, quantum entanglement, non-local games

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.22

1 Introduction

A central concept in theoretical computer science and quantum information is that of a *two-player one-round game*. A two-player game G is specified by question sets X and Y, answer sets A and B, a distribution μ over pairs of questions, and a verification predicate $V : A \times B \times X \times Y \to \{0, 1\}$. The game is played between two cooperating (but non-communicating) players and a referee. The referee samples $(x, y) \in X \times Y$ according to μ

* The author was supported by NSF under CCF-0939370 and CCF-1420956.
† The author was supported by NSF CAREER Grant CCF-1553477, AFOSR YIP award number FA9550-16-1-0495, and the IQIM, an NSF Physics Frontiers Center (NFS Grant PHY-1125565) with support of the Gordon and Betty Moore Foundation (GBMF-12500028).
‡ The author was supported by Simons Foundation grant #360893, and National Science Foundation Grants 1122374 and 1218547.

© Mohammad Bavarian, Thomas Vidick, and Henry Yuen;
licensed under Creative Commons License CC-BY

and sends x and y to each player, who provide answers $a \in A$ and $b \in B$ respectively. The players win the game if their answers satisfy the predicate $V(a, b, x, y)$.

Games arise naturally in settings ranging from hardness of approximation [21] and interactive proof systems [5, 20] to the study of Bell inequalities and non-locality in quantum physics [12, 13]. Depending on the context it is natural to consider players with access to different resources. In particular, in this work we distinguish between the *classical value* $\textsc{Val}(G)$ of a game, defined as the maximum winning probability of players allowed to produce their answers using private and shared randomness, and the *entangled value* $\textsc{Val}^*(G)$, for which the players may use shared entanglement as well .

An important operation on games and the main focus of this work is that of *repeated tensor product* or *parallel repetition*. This operation takes a game G and a parameter m and outputs a new game $G^{\otimes m}$ in which m independent instances of G are simultaneously played with the two players: the referee samples m independent questions $\{(x_i, y_i)\}_{i=1}^m$ from G, sends $(x_1, \ldots, x_m)$ to the first and $(y_1, \ldots, y_m)$ to the second player, and checks their corresponding answers $(a_1, \ldots, a_m)$ and $(b_1, \ldots, b_m)$ using the predicate $V = \prod_{i=1}^m V(a_i, b_i, x_i, y_i)$. Parallel repetition is often used in complexity theory in order to perform some form of *amplification*, such as amplifying the completeness-soundness gap of a proof system. A fundamental question that arises in this context is how the value of a repeated game $G^{\otimes m}$ relates to the value of the original game G and the number of repetitions m.

The behavior of the game value under parallel repetition can in general be quite subtle [19, 18, 31]. In a celebrated paper, Raz showed that if $\textsc{Val}(G) < 1$, then $\textsc{Val}(G^{\otimes m})$ goes to 0 exponentially fast in m [30]. Even with later simplifications and improvements to the proof (e.g. [22, 29, 8]), Raz's parallel repetition theorem remains a substantial technical result.

Recently, Moshkovitz [28] introduced a simple yet powerful framework for parallel repetition, called *parallel repetition via fortification*. In this framework, a game G is transformed through an operation called "fortification" to a new game G'. This new game G' is equivalent to G in that $\textsc{Val}(G) = \textsc{Val}(G')$, but then Moshkovitz shows that behavior of the value of *fortified* games under parallel repetition is much simpler than the general case, and avoids many of the subtleties encountered in the general case. The main benefits of fortified games are two-fold: first, their behavior under parallel repetition is much simpler than the general case, and second, all games can be easily fortified. Thus for nearly all intents and purposes, it suffices to focus on the parallel repetition of fortified games.

Despite its attractive features, the fortification framework [28] has some limitations; for instance it is only applicable to the restricted (though very important) setting of classical two-player projection games. In this work, we continue the study of the fortification approach to parallel repetition and try to expand its scope to wider classes of games.

Previously, an attempt to address the limitation of fortification to projection games was made in [6, Lemma 1.9], but this was only partially successful.[1] Here, using a slightly modified form of the framework (see the discussion of our analytic formulation below), we are able to extend fortification to all *classical games* (i.e. games that are not projection games, and involve any number of players), as well as the challenging setting of *entangled games*.

More precisely, the following is a summary of our main contributions:

[1] The difficulty there was that fortification increases the alphabet size of the game considerably, which in turn increased the additive error of the parallel repetition to the extent that for general two-prover games the whole approach seemed to entirely break down. We get around this difficulty by showing that for the kinds of games (concatenated games) that arise from the fortification procedure it is possible to establish a parallel repetition theorem where the additive error only depends on the alphabet size of the original, rather than the fortified game. See Subsection 1.1 for more on this.

- **Analytic formulation of fortification.** The framework of parallel repetition by fortification was originally cast in combinatorial terms; Moshkovitz's definition of fortified games, which we describe below in detail, involves a guarantee on the value of every sufficiently large rectangular subgame of a game. In our analytic reformulation, fortified games are defined in terms of *substrategies*, which one can think of as randomized strategies for the game where the probability that the players output an answer may be less than 1. This definition behaves much more "smoothly", allowing us to generalize them to the entangled and multiplayer settings.
- **Fortification of general classical games and games with more than two players.** Next, we show how to fortify a general k-player game G, for any $k \geq 2$. We show that for any two fortified general classical games G' and H', $\textsc{Val}(G' \otimes H') \approx \textsc{Val}(G') \cdot \textsc{Val}(H')$. Together this implies new gap amplification results for general (as opposed to projection) two-player and multiplayer classical games.
- **An entangled-value preserving variant of concatenation.** A major obstacle in extending the fortification framework to the quantum setting is that concatenation, the main ingredient of the original fortification results, does not in general preserve the entangled value. That is, if G' is the fortification of G, it doesn't generally hold that $\textsc{Val}^*(G') = \textsc{Val}^*(G)$ (even though $\textsc{Val}(G') = \textsc{Val}(G)$). This is problematic for obtaining gap amplification results: if $\textsc{Val}^*(G) = 1$, then $\textsc{Val}^*(G^{\otimes n}) = 1$, but $\textsc{Val}^*(G'^{\otimes n})$ could be exponentially small!
  To resolve this issue, we augment the ordinary concatenation procedure of [28] by giving the players some auxiliary advice input (see Definition 4) which helps in keeping the entangled value unchanged. Using this, we define a variant of the fortification transformation which we call *ordered fortification*. As desired, in addition to preserving the classical value, this transformation also preserves the entangled value, which is essential for the completeness of our gap amplification result.
- **Fortification of games with entangled players.** We show that for a general two-player game G, its ordered-fortification G_{OF} is a two-player game such that $\textsc{Val}^*(G_{OF}) = \textsc{Val}^*(G)$, and is also quantumly fortified. We then prove that for any two quantumly fortified games G' and H', $\textsc{Val}^*(G' \otimes H') \approx \textsc{Val}^*(G') \cdot \textsc{Val}^*(H')$. Together this implies a new general gap amplification method for entangled two-player games. This (see Theorem 3) is the most technically challenging component of this work.

Let us note that our extensions of the fortification approach, as described in the last three items above, are ultimately enabled by the analytic viewpoint described in point 1. In order to describe the main ideas behind these results, we first briefly recall the combinatorial framework from [28].

The combinatorial framework

Let G be a two-player game with question sets X, Y and acceptance predicate V. For $S \subseteq X$ and $T \subseteq Y$, the *subgame* $G_{S\times T}$ is defined as the game where the referee selects $(x, y) \in X \times Y$ according to μ conditioned on $x \in S, y \in T$ and checks the players' answers according to the same predicate V (the referee accepts automatically if $\mu(S \times T) = 0$). A game G is said to be (ε, δ)-*combinatorially fortified*[2] if

$$\textsc{Val}(G_{S\times T}) \leq \textsc{Val}(G) + \varepsilon, \qquad \forall\, S \subseteq X,\, T \subseteq Y,\ \text{s.t. } \mu(S \times T) \geq \delta. \tag{1}$$

[2] We shall refer to this notion as combinatorial fortification to distinguish it from the distinct (though closely related) notion of *analytic fortification* we primarily use throughout the paper; see Definition 19.

The main insight underlying [28] is that games satisfying (1) also satisfy a strong form of parallel repetition (up to some number of rounds depending on ε, δ, and the alphabet size of G). This motivates the following approach to parallel repetition: Given a game G, Moshkovitz transforms the game $G \to G'$ such that $\textsc{Val}(G') \approx \textsc{Val}(G)$ and G' is (ε, δ)-combinatorially fortified for an appropriate choice of (ε, δ). Since fortified games satisfy a strong form of parallel repetition, one expects

$$\textsc{Val}(G'^{\otimes m}) \approx \textsc{Val}(G')^m \approx \textsc{Val}(G)^m. \tag{2}$$

Indeed, by appropriately choosing the parameters (ε, δ), [28] can show that the full procedure

$$G \longrightarrow G' \longrightarrow G'^{\otimes m} \tag{3}$$

amounts to a size-efficient method of *gap amplification*. That is, we have

$$\begin{array}{lcl} \textsc{Val}(G) \geq c & \Rightarrow & \textsc{Val}(G'^{\otimes m}) \gtrsim c^m \\ \textsc{Val}(G) \leq s & \Rightarrow & \textsc{Val}(G'^{\otimes m}) \lesssim s^m \end{array}, \tag{4}$$

where we refer to the first condition as completeness and the second as soundness. The gap amplification procedure of Moshkovitz $G \to G' \to G'^{\otimes m}$ from (3) has three components: (i) a preprocessing step (biregularization), (ii) fortification, (iii) parallel repetition for fortified games.

The goal of the preprocessing step – the simplest step of the three – is to make the game *biregular* (a game G is called biregular if the marginals of questions on both Alice and Bob sides are uniform), since it is typically easier to analyze the fortification procedure for such games. The second step is *fortification*, which is the main technical ingredient of the whole approach. It is achieved by "concatenating" the game (see Section 1.1 below) with appropriate bipartite pseudorandom graphs. The third step $G' \to G'^{\otimes m}$ is the parallel repetition of fortified games, which as observed by [28] is considerably simpler to analyze than the general (non-fortified) games.

1.1 Results and techniques

The main result of our work is the extension of the fortification framework to general classical games (with any number of players) and two-player entangled games. On the way to these results we prove new results on all three components of the fortification framework: (i) biregularization, (ii) fortification, and (iii) parallel repetition. In this subsection, we discuss some of these results in detail.

Parallel repetition

A main contribution of [28] was the realization that the two-player projection fortified games satisfy a strong form of parallel repetition, up to an additive error. This additive error depended on the parameters of fortification as well as the alphabet size of the fortified game. In this work, we prove an improved parallel repetition theorem (Theorem 22) which has the same dependance in the parameters of fortification, but instead of the alphabet size of the resulting fortified game, it only depends on the alphabet size of the original game (which has exponentially smaller alphabet size). This new parallel repetition theorem is crucial for extending the fortification framework to the setting of general (as opposed to projection) two-player games.

Let us remark that the reason why alphabet blow-up of fortification does not cause an issue for projection games is because for projection games it suffices to only fortify one side of the game (by working with so-called "square projection" version of the game). As a result there is no alphabet blow-up for the "unfortified" side, which allows the arguments of [28, 6] to go through. This one-sided fortification does not work for general games, which is why we need Theorem 22.

Fortification

We start with a definition. Let $G = (X \times Y, A \times B, \mu, V)$ be a game, and M and P two bipartite graphs over vertex sets (X', X) and (Y', Y) respectively. For each $x \in X$ or $x' \in X'$ let $N(x) \subseteq X'$ and $N(x') \subseteq X$ denote the set of neighbors of x and x', respectively (similarly for any y, y').

▶ **Definition 1** (Concatenated game [28])**.** In the concatenated game $G' = (M \circ G \circ P)$, the referee selects questions (x, y) according to μ, and independently selects a random neighbor x' for x using M, and y' for y using P. The players receive questions x' and y' and respond with assignments $a' : N(x') \to A$ and $b' : N(y') \to B$ respectively. The players win if $V(a'(x), b'(y), x, y) = 1$.

Our first two main results show how, both in the classical and quantum settings, any game can be fortified by concatenating it with bipartite graphs M and P with sufficiently good spectral expansion.[3] (See Section 2.4 for the definition of spectral expanders, and Section 3.1 for the notion of weak fortification.)

▶ **Theorem 2** (Main-classical)**.** *Let G be a biregular game and M and P two bipartite λ-spectral expanders. If $\lambda \leq \frac{\varepsilon}{2}\sqrt{\frac{\delta}{2}}$, then the concatenated game $G' = (M \circ G \circ P)$ is (ε, δ)-weakly fortified against classical substrategies.*

▶ **Theorem 3** (Main-quantum)**.** *Let G be a biregular game and M and P two bipartite λ-spectral expanders. If $\lambda \leq \frac{\varepsilon^2\delta}{56}$, then the concatenated game $G' = (M \circ G \circ P)$ is (ε, δ)-weakly fortified against entangled strategies.*

We stress that both in the quantum and classical settings the procedure used to fortify a game is precisely the same, i.e. concatenation with spectral expanders, and the only difference is in the resulting parameters. Despite the similarities, the proof of Theorem 3 is significantly more involved, requiring several new ideas and substantial matrix analytic arguements.

Next we discuss a distinctively quantum phenomenon which makes the construction of a full quantum gap amplification theorem – quantum analogue of (4) – considerably more difficult. As it turns out, even though Theorem 3 is sufficient to prove the soundness case of the gap amplification theorem, the concatenation procedure used in the process can undermine the completeness condition (i.e. $\textsc{Val}^*(G'^{\otimes m}) \gtrsim \textsc{Val}^*(G)^m$ in general fails to hold).

The issue is as follows: let G be a game and $G' = (M \circ G \circ P)$ be a concatenated version of G. Classically we have $\textsc{Val}(G') = \textsc{Val}(G)$. Quantumly, even though we still have $\textsc{Val}^*(G') \leq \textsc{Val}^*(G)$ the other direction in general fails: we would have liked to argue that the players in G' are able to utilize the strategy in G to achieve the same success probability

[3] Even though explaining these results in full requires the definition of analytically fortified games, which we introduce only later in Section 3.1, the analogy with the notion of combinatorially fortified games from (1) should still be sufficient to understand the basic ideas.

in the concatenated game, but this seems impossible: having received $x' \in X'$ and $y' \in Y'$, the players have access to lists $N(x') \subseteq X$ and $N(y') \subseteq Y$ that they know contain the true questions of the referee, i.e. $x^* \in N(x')$, $y^* \in N(y')$. The players would like to apply their optimal strategy in G to each and every $(x, y) \in N(x') \times N(y')$ simultaneously, but this is in general impossible in the quantum setting.[4]

Note that the same issue does not arises classically because the optimal strategy in G can be taken to be a deterministic one, and the players in G' can use the same labeling suggested by the optimal strategy in G to give labels to all of $N(x')$ and $N(y')$ simultaneously. This strategy however relies on the fact that classically different questions have a simultaneous labeling, a fact which certainly has no quantum analogue.

We resolve the above issue using a novel entangled value-preserving variant of fortification which we call *ordered fortification*. The basic idea for ordered fortification is to give the players some extra advice information which helps in preserving the entangled value.

Let G be a game and $G' = (M \circ G \circ P)$ be a concatenated version of G. There is an extra parameter l in the construction defined as $l = \max\{\max_{x' \in X'} |N(x')|, \max_{y' \in Y'} |N(y')|\}$.

▶ **Definition 4** (Ordered concatenation)**.** Let G and G' be as above. In G'_{OF}, the referee samples (x, y) according to G and picks random neighbors $x' \sim N(x)$ and $y' \sim N(y)$ independently. She then also picks two random injective maps $r_{x'} : N(x') \to [l]$ and $s_{y'} : N(y') \to [l]$ conditioned on $s_{x'}(x) = r_{y'}(y)$. The referee sends x' and $r_{x'}$ to the first player, and y' and $s_{y'}$ to the second and accepts if the players' answers $a' : N(x') \to A$ and $b' : N(y') \to B$ satisfy $V(a'(x), b'(y), x, y) = 1$.

Here the crucial point is that $r_{x'}$ and $s_{y'}$ are correlated. They give matching labels to true questions x and y. To achieve the same winning probability as in G, the players in G'_{OF} will share l copies of the state $|\psi\rangle$ from the optimal strategy in G. For each $x^* \in N(x')$ with label $i = r_{x'}(x^*)$, the first player will apply the optimal G-strategy for x to the i^{th} copy of $|\psi\rangle$ (similarly for the second player). The fact that $r_{x'}(x) = s_{y'}(y)$ ensures that for the true questions x and y the players apply the optimal G strategies to the same copy of $|\psi\rangle$, and hence are able to win with exactly the same winning probability as in G.

Of course, the crucial part here is that even though the auxiliary information in $r_{x'}$ and $s_{y'}$ is helpful to the players for replicating the winning probability of G, it should not be "too helpful". In particular, we need to still be able to prove that G'_{OF} is fortified with appropriate parameters. This point is established by the following theorem.

▶ **Theorem 5** (Main-ordered fortification)**.** *Let G be a game and M and P be two bipartite graphs as above. Let G'_{OF} be constructed from G and $G' = (M \circ G \circ P)$ as in Definition 4. Then, we have*

$$\textsc{Val}^*(G'_{OF}) = \textsc{Val}^*(G).$$

Furthermore if M and P are λ-spectral expanders and $\lambda \leq \frac{\varepsilon^2\delta}{56}$, then G'_{OF} is also (ε, δ) weakly fortified.

We prove Theorem 5 in Section 6 using a spectral argument that reduces it to Theorem 3. Beside the above, we also prove a simple multiplayer fortification in Section 5 for classical

[4] This is because the measurement operators of different questions do not in general commute which prevents Alice (say) to obtain simultaneous answers for all questions in $N(x')$. As a further illustration of this issue, see Section 2.2 for an example of a game where $\textsc{Val}^*(G') < \textsc{Val}^*(G)$.

games. It may be possible to adapt the proofs of Theorem 3 and Theorem 28 to obtain a *multiplayer fortification theorem* for entangled games. Although plausible, some further technical issues arise in this case which we do not pursue here.

Biregularization

As already mentioned, biregularization is a minor (but necessary) step in the fortification framework. Our biregularization lemmas are presented in Subsection 2.3 and are proved in Appendix A. In terms of final statement, our biregularization lemmas are incomparable with those of [28, 6]. For example, in the case of graphical games, we prove a biregularization lemma which preserves the value exactly but has a cubic blow-up in the number of questions, whereas the biregularization lemmas from [6, 28] had a nearly linear blow-up but only preserved value up to an additive error. Moreover, in this work we prove biregularization for all games whereas [6, 28] only considered graphical games. (See Subsection 2.3 for definitions.)

1.2 Related work

The main result underlying the present work is Moshkovitz [28], where the framework of parallel repetition via fortification was first introduced. Some simplifications and corrections to the work of Moshkovitz appeared in Bhangale et al. [6]. In particular, an important contribution of [6] was the clarification of the best bounds possible in classical fortification theorems [6, Appendix C]. Going back, the general idea of modifying the game in order to facilitate its analysis under parallel repetition originates from the work of Feige and Kilian [17] who introduced the confuse/miss-match style repetition of games. The Feige-Kilian type parallel repetition was later extended by Kempe and Vidick [25] to the quantum setting allowing them to obtain the first general parallel repetition theorem for quantum games.[5]

Another important set of ideas underlying our work is related to the analytic approach to parallel repetition pioneered by Dinur and Steurer [15], further extended by Dinur et al. [16]. Our analytic reformulation of fortification framework is very much inspired by the ideas in these works.

Yet another different stream of work (more distantly related to the present work) follows the original ideas of Raz and Holenstein [30, 22] by taking a more information theoretic approach to quantum parallel repetition. The first results in this direction were obtained by Chailloux and Scarpa [10] and Jain et al. [24] who prove exponential-decay parallel repetition results for free two-player games. Their analysis, as well as the follow-up work of Chung et al. [11], provided the basis for the recent work of the authors [4] who obtained hardness amplification method in great generality by introducing and analyzing the parallel repetition of a class of games called *anchored games*. The final hardness amplification results obtained here through fortification are incomparable to that of [4]: fortification allows for a somewhat faster rate of decay in some regimes,[6] yet it suffers from a much larger blow-up in terms of alphabet size.

[5] Their transformation did not preserve the entangled value, and hence did not lead to a fully general hardness amplification method for entangled games (being restricted to the case where the completeness holds with classical strategies). Our Theorem 3, without the improvement of Theorem 5, is applicable to a similar setting.

[6] For example, for general two-player games the fortification approach gives a nearly perfect decay in terms of number of repetitions and question size blow-up, whereas [4] and other information theoretic results have a $(1-\varepsilon^2)^{\Omega(m)}$ type behavior which is weaker than fortification when $m \ll \varepsilon^{-1}$.

Turning to the multiplayer setting, very little was known prior to the present work and [4]. It is folklore that free games with any number of players satisfy a parallel repetition theorem, and this was explicitly proved in both classical and quantum settings in [11]. Multiplayer parallel repetition has been studied in the setting of *non-signaling strategies*, a superset of entangled strategies which allows the players to generate any correlations that do not imply communication. Buhrman et al. [9] show that the non-signaling value of a game G with any number of players decays exponentially under parallel repetition, with a rate of decay that depends on the entire description of the game G. Arnon-Friedman et al. [1] and Lancien and Winter [27] achieve similar results using a different technique based on "de Finetti reductions". An interesting fact about the latter work [27] is the use of a notion of sub-no-signalling strategies which seems related to our notion of quantum/classical substrategies.

1.3 Organization

In Section 2, we introduce some basic definitions and notation including the notion of substrategies, induced strategies, and some other basic results and definitions that are used throughout the paper. In Section 3, we complete the presentation of our main results (which was started in the introduction), discuss the parameters of the final gap amplification results, and present the formal definition of analytically fortified games.

The remaining sections contain proof of the main theorems. Our parallel repetition theorem is proved in Section 4. Theorems 2 and 3 are proved in Sections 5 and 7, respectively. The reduction from Theorem 5 to Theorem 3 is given in Section 6. The biregularization lemmas are proved in Appendix A. We conclude by some open problems in Section 8.

2 Preliminaries

Given a distribution μ, by $z \sim \mu$ we mean that the random variable z is distributed according to μ. For a set S, by $z \sim S$ we mean $z \sim U_S$ where U_S is the uniform distribution over S. For Hermitian matrices A, B we write $A \geq B$ if and only if $A - B$ is positive semidefinite.

Given two games G_1 and G_2, we define the tensor product game $G_1 \otimes G_2$ as the game where the referee selects two pairs of questions (x_1, y_1) and (x_2, y_2) independently according to G_1 and G_2, sends (x_1, x_2) and (y_1, y_2) to Alice and Bob respectively, and checks their answers (a_1, a_2) and (b_1, b_2) according to the product predicate $\prod_{i=1}^{2} V(a_i, b_i, x_i, y_i)$.

2.1 Game value and strategies

The main goal of this section is to introduce the notion of classical and quantum *substrategies* which replace the notion of *subgames* from [28, 6]. As subgames were central in the *combinatorial* framework of [28], substrategies are similarly central to our analytic framework.

Let G be a game with question sets X, Y, answer sets A, B, predicate V, and question distribution μ on $X \times Y$.

▶ **Definition 6** (Classical substrategies)**.** Let $G = (X \times Y, A \times B, \mu, V)$ be a two-player game. A *classical substrategy* is given by (f, g) where $f : X \times A \to [0, 1], g : Y \times B \to [0, 1]$ satisfy

$$\forall x \in X,\ f(x) := \sum_a f(x, a) \leq 1, \qquad \forall y \in Y,\ g(y) := \sum_b g(y, b) \leq 1.$$

We call (f, g) a *"complete strategy"* (sometimes simply *strategy*) if equality holds in all above inequalities, i.e. $f(x) = g(y) = 1$ for all x, y.

▶ **Definition 7.** Given a substrategy (f, g), the value of G with respect to (f, g) is given by

$$\textsc{Val}(G, f, g) := \mathop{\mathbb{E}}_{(x,y)\sim\mu} \sum_{a\in A, b\in B} V(a, b, x, y)\, f(x, a) \cdot g(y, b). \tag{5}$$

The *classical value* of G is

$$\textsc{Val}(G) := \sup_{f,g} \textsc{Val}(G, f, g), \tag{6}$$

where the supremum is taken over all complete strategies f, g.

We note that the definition given by (6) can be easily seen to be equivalent to the more traditional definition of the classical value, i.e.

$$\textsc{Val}(G) := \max_{\substack{p:X\to A\\ q:Y\to B}} \mathop{\mathbb{E}}_{(x,y)\sim\mu} V(p(x), q(y), x, y), \tag{7}$$

because any strategy $f : X \times A \to [0, 1], g : Y \times B \to [0, 1]$ can be written as convex combination of a collection of strategies of $\{0, 1\}$ valued strategies; on the other hand, taking supremum over f, g which are $\{0, 1\}$ valued is precisely equivalent to (7).

Next, we extend the above notions to the quantum setting.

▶ **Definition 8** (Quantum substrategies)**.** Let $G = (X \times Y, A \times B, \mu, V)$ be a two-player game. A *quantum* (or *entangled*) *substrategy* for G is a tuple $(|\psi\rangle, \{A_x^a\}, \{B_y^b\})$ defined by an integer $d \in \mathbb{N}$, a unit vector $|\psi\rangle \in \mathbb{C}^{d\times d}$ and sets of positive semi-definite matrices $\{A_x^a\}_{x\in X, a\in A}, \{B_y^b\}_{y\in Y, b\in B}$ over $\mathbb{C}^d$ satisfying

$$\forall x \in X,\ A_x := \sum_a A_x^a \le \mathrm{Id}, \qquad \forall y \in Y,\ B_y := \sum_b B_y^b \le \mathrm{Id}\,. \tag{8}$$

If $A_x = B_y = \mathrm{Id}$ for every x, y the quantum substrategy is called a *"complete strategy"* (sometimes simply *strategy*).

▶ **Definition 9.** Given a quantum substrategy $(|\psi\rangle, \{A_x^a\}, \{B_y^b\})$, the value of G with respect to $(|\psi\rangle, \{A_x^a\}, \{B_y^b\})$ is given by

$$\textsc{Val}^*(G, |\psi\rangle, \{A_x^a\}, \{B_y^b\}) = \mathop{\mathbb{E}}_{(x,y)\sim\mu} \sum_{a,b} V(a, b, x, y) \langle\psi| A_x^a \otimes B_y^b |\psi\rangle.$$

The *entangled value* of G is defined as

$$\textsc{Val}^*(G) = \sup_{|\psi\rangle, \{A_x^a\}, \{B_y^b\}} \textsc{Val}^*(G, |\psi\rangle, \{A_x^a\}, \{B_y^b\}), \tag{9}$$

where the supremum is taken over all complete strategies $(|\psi\rangle, \{A_x^a\}, \{B_y^b\})$.

2.2 Concatenated games

Let $G = (X \times Y, A \times B, \mu, V)$ be a game, and M and P two bipartite graphs over vertex sets (X', X) and (Y', Y) respectively. For each $x \in X$ or $x' \in X'$ let $N(x) \subseteq X'$ and $N(x') \subseteq X$ denote the set of neighbors of x and x', respectively (similarly for any y, y'). Recall the definition of Concatenated Games from the introduction.

▶ **Definition 10** (Definition 1 restated)**.** In the concatenated game $G' = (M \circ G \circ P)$, the referee selects questions (x, y) according to μ, and independently selects a random neighbor x' for x using M, and y' for y using P. The players receive questions x' and y' and respond by assignments $a' : N(x') \to A$ and $b' : N(y') \to B$ respectively. The players win if $V(a'(x), b'(y), x, y) = 1$.

For a concatenated game $G' = (M \circ G \circ P)$, we refer to G' as the *outer game* and to G as the *inner game*.

Let $G' = (M \circ G \circ P)$ be a concatenated game. Let $d_{X'} = \max_{x' \in X'} |N(x')|$, $d_{Y'} = \max_{y' \in B'} |N(y')|$. Then, the alphabet of the concatenated game is given by $A' = A^{d_{X'}}$, $B' = B^{d_{Y'}}$. Similarly, it is easy to see that the distribution μ' of questions in G' is given by $\mu'(x', y') = \mathbb{E}_{(x,y)\sim\mu} \frac{1_{x' \in N(x)}}{|N(x)|} \cdot \frac{1_{y' \in N(y)}}{|N(y)|}$.

▶ **Definition 11.** Let $G' = (M \circ G \circ P)$ be a concatenated game. To any pair of substrategies (f, g) for G' we associate the *induced substrategy*[7]

$$f(x, a) := \mathop{\mathbb{E}}_{x' \sim N(x)} \sum_{a' : a'(x) = a} f(x', a'), \qquad g(y, b) := \mathop{\mathbb{E}}_{y' \sim N(y)} \sum_{b' : b'(y) = b} f(y', b'). \tag{10}$$

Similarly, given an entangled substrategy $(|\psi\rangle, \{A^{a'}_{x'}\}, \{B^{b'}_{y'}\})$ for G', we define the *induced substrategy* as

$$A^a_x := \mathop{\mathbb{E}}_{x' \sim N(x)} \sum_{a'(x) = a} A^{a'}_{x'}, \qquad B^b_y := \mathop{\mathbb{E}}_{y' \sim N(y)} \sum_{b'(y) = b} B^{b'}_{y'}. \tag{11}$$

Intuitively, an induced strategy is a strategy for the inner game in which the players proceed as follows: given question $x \in X$, $y \in Y$ and a strategy (f, g) for the outer game, the players select two random neighbors of their questions $x' \in N(x), y' \in N(y)$ independently, and play according to the labeling of x, y suggested by (f, g) at x' and y'.

The following simple proposition will play an important role throughout the paper.

▶ **Proposition 12.** *Let $G' = (M \circ G \circ P)$ be a concatenated game. The value of any classical strategy (f, g) (resp. quantum strategy $(|\psi\rangle, \{A^{a'}_{x'}\}, \{B^{b'}_{y'}\})$) in the outer game G' is equal to the value of the induced strategy in the inner game G:*

$$\textsc{Val}(G', f, g) = \textsc{Val}(G, f, g) \tag{12}$$

$$\textsc{Val}^*(G', |\psi\rangle, \{A^{a'}_{x'}\}, \{B^{b'}_{y'}\}) = \textsc{Val}^*(G, |\psi\rangle, \{A^a_x\}, \{B^b_y\}). \tag{13}$$

As a consequence,

$$\textsc{Val}(G') \leq \textsc{Val}(G), \qquad \textit{and} \qquad \textsc{Val}^*(G') \leq \textsc{Val}^*(G). \tag{14}$$

Furthermore,

$$\textsc{Val}(G') = \textsc{Val}(G). \tag{15}$$

[7] Note the slight (but convenient) abuse of notation due to the use of the same letter to represent a substrategy and the corresponding induced substrategy. The more accurate but more cumbersome way of denoting the induced strategies in in [15]'s language would have been Mf and Pg.

Proof. The first equality in (12) follows from linearity of expectation and the definition of induced strategies as

$$\begin{aligned}\textsc{Val}(G', f, g) &= \mathop{\mathbb{E}}_{(x,y)\sim\mu} \mathop{\mathbb{E}}_{x'\sim N(x)} \mathop{\mathbb{E}}_{y'\sim N(y)} \sum_{a'\in A', b'\in B'} V(a'(x), b'(y), x, y)\, f(x', a') \cdot g(y', b') \\ &= \mathop{\mathbb{E}}_{(x,y)\sim\mu} \sum_{a\in A, b\in B} V(a, b, x, y)\, f(x, a) \cdot g(y, b) \\ &= \textsc{Val}(G, f, g).\end{aligned}$$

The second equality is proved similarly. The two inequalities (14) follow directly from (12). To show (15) it remains to show that $\textsc{Val}(G') \geq \textsc{Val}(G)$. Consider an optimal deterministic strategy for G given by $p : X \to A$ and $q : Y \to B$. For any $x' \in X'$, $y' \in Y'$ define $a' : N(x') \to A$ according to p and $b' : N(y') \to B$ according to q. It is easy to see that this achieves the same value in G' as (p, q) did in G. ◀

As mentioned in the introduction, the quantum analogue of (15) does not hold in general. For example, consider the case that M and P are complete bipartite graphs. In this case, the players playing $G' = (M \circ G \circ P)$ need to provide a labeling to all vertices in X and Y simultaneously. But this is essentially just a classical strategy as the labelings for X, Y are now fixed. Hence, $\textsc{Val}^*(G') = \textsc{Val}(G)$, the classical value, which in many cases could be much smaller than $\textsc{Val}^*(G)$.

2.3 Biregularization

As in [28, 6] we prove our fortification theorems for the special class of *biregular games*.

▶ **Definition 13.** A two-prover game $G = (X \times Y, A \times B, \mu, V)$ is called biregular if the marginals of μ on X and Y are both uniform.

The following lemma justifies that for our purposes we may always assume a game is biregular.

▶ **Lemma 14** (Biregularization lemma)**.** *Let $G = (X \times Y, A \times B, \mu, V)$ be a two-prover game and $\tau \in (0, 1)$ a fixed constant. There exists an efficient algorithm that given G produces a biregular game G_{int} with question sets X_{int} and Y_{int} of cardinality at most*

$$|X_{int}| \leq \frac{8|X|^2|Y|}{\tau}, \qquad |Y_{int}| \leq \frac{8|X||Y|^2}{\tau}, \tag{16}$$

the same answer alphabet size as G, and value satisfying

$$\textsc{Val}(G) \leq \textsc{Val}(G_{int}) \leq \textsc{Val}(G) + \tau, \qquad \textsc{Val}^*(G) \leq \textsc{Val}^*(G_{int}) \leq \textsc{Val}^*(G) + \tau. \tag{17}$$

Note that (17) implies that applying the Biregularization Lemma to a game never decreases its value, and hence the procedure is completeness preserving.

A widely used class of games in applications are so-called *graphical games*, for which we can get an improved biregularization result that does not require any approximation factor τ.

▶ **Definition 15.** A graphical game G is a game where the questions are given by choosing an edge of a bipartite graph uniformly at random (i.e. $E \subseteq X \times Y$ and $\mu(x, y) = \frac{1}{|E|}$ if $(x, y) \in E$ and $\mu(x, y) = 0$ otherwise). The predicate and the answers do not have any restrictions.

▶ **Lemma 16** (Biregularization lemma, graphical case)**.** *Suppose G is two-prover graphical game with E edges between (X, Y). There exists an efficient algorithm that given G produces a biregular game G_{int} with question sets X_{int} and Y_{int} bounded by*

$$|X_{int}| \leq |E| \cdot |X| \leq |X|^2|Y|, \qquad |Y_{int}| \leq |E| \cdot |Y| \leq |X||Y|^2, \tag{18}$$

the same answer alphabet size as G, and the value satisfying

$$\textsc{Val}(G) = \textsc{Val}(G_{int}) \qquad \textsc{Val}^*(G) = \textsc{Val}^*(G_{int}). \tag{19}$$

▶ Remark. In the above, we can allow for multiple edges across vertices of G. In this case E must be taken as a multi-set and the bound $|E| \leq |X||Y|$ used in (18) must be suitably modified.

Interestingly, our technique for proving the biregularization lemmas is concatenation itself! This is done in Appendix A.

2.4 Expanders

The method used in [28, 6] for fortifying a game is concatenation with sufficient pseudorandom bipartite graphs. This is done using extractors in [28] whereas expanders are employed in [6].[8] Here we follow the latter approach and use expanders.

Let $M = (X' \times X, E)$ be a bipartite graph. For $x \in X$ let $N(x) \subseteq X'$ denote the set of neighbors of x and similarly for $x' \in X'$. We shall work with graphs that are X-regular, i.e. $d = |N(x)|$ for all $x \in X$. Define distributions μ and μ' on X and X' via

$$\mu(x) = \frac{1}{|X|}, \qquad \mu'(x') = \frac{|N(x')|}{d}.$$

for all $x \in X$ and $x' \in X'$. Note that $\mu'(x')$ is the probability of obtaining x' by sampling $x \sim \mu$ and taking a random neighbor of (according M) x . Let $\mathcal{M}$ be the following normalized adjacency matrix of M

$$\mathcal{M}(x, x') = \begin{cases} \frac{1}{d} \cdot \sqrt{\frac{\mu(x)}{\mu'(x')}} & \text{if } x' \in N(x) \\ 0 & \text{otherwise} \end{cases}$$

We usually view $\mathcal{M}$ as an operator from $\ell_2(X')$ to $\ell_2(X)$. Note that when M is a biregular expander we get the simpler definition $\mathcal{M}(x, x') = \frac{1}{d}\sqrt{\frac{|X'|}{|X|}}$ for $x' \in N(x)$, and 0 otherwise.

▶ **Definition 17.** A bipartite graph M is called a λ-*spectral expander* if the second-largest singular value of $\mathcal{M}$ is at most λ.

A simple useful proposition for us is the following:

▶ **Proposition 18.** *Let $M = (X' \times X, E)$ be a bipartite λ-spectral expander. For $f : X' \to \mathbb{R}$ and $x \in X$ let $f(x) = \mathbb{E}_{x' \sim N(x)} f(x')$, and $\bar{f} = \mathbb{E}_{x' \sim \mu'} f(x') = \mathbb{E}_{x \sim \mu} f(x)$. Then*

$$\mathop{\mathbb{E}}_{x \sim \mu} (f(x) - \bar{f})^2 \leq \lambda^2 \mathop{\mathbb{E}}_{x' \sim \mu'} (f(x') - \bar{f})^2. \tag{20}$$

[8] The two approaches however lead to essentially to similar parameters (e.g. $\lambda = O(\varepsilon\sqrt{\delta})$ to get (ε, δ)-fortified graph where λ is the second largest singular value of normalized adjacency matrix of the concatenating graph.); moreover, in the classical setting the approaches are in fact are more or less equivalent. See [6] for more.

Proof. Let $p_\mu \in \mathbb{R}^X, p_{\mu'} \in \mathbb{R}^{X'}$ to unit vectors defined as $p_\mu(x) := \sqrt{\mu(x)}$ and $p_{\mu'}(x') := \sqrt{\mu(x')}$. Let$q_X(x) := \sqrt{\mu(x)}\, f(x)$, $q_{X'}(x') := \sqrt{\mu(x')}\, f(x')$.

First, observe that $\mathcal{M}\, p_{\mu'} = p_\mu$ and $\mathcal{M}^t\, p_{\mu'} = p_\mu$. It follows that $(p_\mu, p_{\mu'})$ form a pair of singular vectors of $\mathcal{M}$. Moreover, it is easy to see[9] that these are top singular vectors which shows that $\|\mathcal{M}\|_{op} = 1$. Now notice that

$$\mathop{\mathbb{E}}_{x' \leftarrow \mu'} (f(x') - \bar{f})^2 = \sum_{x'} (\sqrt{\mu(x')} f(x') - \bar{f}\sqrt{\mu(x')})^2 = \|q_{X'} - \bar{f} p_{\mu'}\|_2^2, \tag{21}$$

Second, observe

$$\mathcal{M}\, q_{X'} = q_X. \tag{22}$$

As such, (20) precisely corresponds to

$$\|q_X - \bar{f}\, p_\mu\|_2^2 = \|\mathcal{M}\,(q_{X'} - \bar{f}\, p_{\mu'})\|_2^2 \le \lambda^2 \cdot \|q_{X'} - \bar{f}\, p_{\mu'}\|_2^2. \tag{23}$$

The claim follows by noting the orthogonality property

$$\langle q_{X'} - \bar{f}\, p_{\mu'}, p_{\mu'} \rangle = \sum_{x'} \mu(x') f(x') - \bar{f} = 0. \tag{24}$$

◀

3 Fortification Framework

This section introduces the fortification framework. We define the notion of analytically fortified games and recall our main parallel repetition and fortification theorems. We end by a discussion of the parameters of the resulting gap amplification results.

3.1 Analytical fortification

We distinguish between two variants of the notion of fortified games which we call *weakly fortified games* and *strongly fortified games*. Although the difference between the two may seem minor, this difference is in fact quite important in the quantum case.

▶ **Definition 19** (Fortified games)**.** Let $\varepsilon, \delta \in [0,1]$. A concatenated game $G' = (M \circ G \circ P)$ is called weakly (ε, δ)- fortified against classical substragies if for any substrategy f, g we have

$$\textsc{Val}(G', f, g) \le (\textsc{Val}(G) + \varepsilon) \cdot \mathop{\mathbb{E}}_{(x,y)\sim\mu} f(x)\, g(y) + \delta. \tag{25}$$

Similarly, we define G' to be weakly (ε, δ)-fortified against entangled substrategies if for any substrategy $\{A_{x'}^{a'}\}, \{B_{y'}^{b'}\}$ we have

$$\textsc{Val}^*(G', \{A_{x'}^{a'}\}, \{B_{y'}^{b'}\}) \le (\textsc{Val}^*(G) + \varepsilon) \cdot \mathop{\mathbb{E}}_{(x,y)\sim\mu} \langle \psi | A_x \otimes B_y | \psi \rangle + \delta. \tag{26}$$

If furthermore $\textsc{Val}(G)$ (resp. $\textsc{Val}^*(G)$) can be replaced by $\textsc{Val}(G')$, (resp. $\textsc{Val}^*(G')$) in the above then the game is called "strongly fortified" against classical (resp. quantum) substrategies.

[9] e.g. by appealing to the Perron-Frobenius theorem.

Note that our main results, Theorems 2 and 3, show how any game can be (weakly) fortified by concatenating it with good-enough spectral expanders.

Two remarks regarding the above definition are in order:

- Using (14), we see that strong fortification implies weak fortification, as expected from the terminaology.
- From (15) it follows that the two notions in fact coincide in the case of classical fortification, but this is no longer the case for quantum fortification.

Our notion of fortified games and that of [28, 6] are closely related. Essentially, in Definition 19 we have replaced the the condition for all δ-large rectangles in (1) with a smoother condition. In terms of a precise relation, we can show the following.

▶ **Claim 20.** *Every $(\varepsilon, \varepsilon\delta)$ strongly fortified game is also $(2\varepsilon, \delta)$ combinatorially fortified.*

Proof. Consider a subgame given by $S \subseteq X, T \subseteq Y$ in G. To every strategy (p, q) for $G_{S\times T}$, i.e., $p : S \to A$, $q : T \to B$, we can associate a natural substrategy (f, g) by

$$f(x,a) = \begin{cases} 1 & \text{if } x \in S \wedge p(x) = a, \\ 0 & \text{otherwise} \end{cases}, \qquad g(y,b) = \begin{cases} 1 & \text{if } y \in T \wedge q(y) = b, \\ 0 & \text{otherwise} \end{cases}. \tag{27}$$

Then one can easily see

$$\textsc{Val}(G, f, g) = \textsc{Val}(G_{S\times T}, p, q) \cdot \mu(S \times T).^{10} \tag{28}$$

Now assuming that rectangle $S \times T$ is δ-large, i.e. $\mu(S \times T) \geq \delta$, and since G is fortified against classical substrategies, we have

$$\begin{aligned} \textsc{Val}(G_{S\times T}, p, q) &= \frac{\textsc{Val}(G, f, g)}{\mu(S \times T)} && (29)\\ &\leq (\textsc{Val}(G) + \varepsilon) \cdot \frac{\mathbb{E}_{(x,y)\sim\mu} f(x)g(y)}{\mu(S \times T)} + \frac{\delta\varepsilon}{\mu(S \times T)} && (30)\\ &\leq \textsc{Val}(G) + 2\varepsilon && (31) \end{aligned}$$

where in the second inequality we used $\mu(S \times T) = \mathbb{E}_{(x,y)\sim\mu} f(x)g(y)$. ◀

We note that in Lemma 20, the reverse implication does not hold and the notion of analytically fortified game is strictly stronger. In what follows, in the rare occasion when we call a game fortified (without specifying weak or strong) we mean strongly fortified.

3.2 Parallel repetition of fortified games

Using the definition of fortified games, it is straightforward to prove the following parallel repetition theorem.

▶ **Theorem 21** (Basic parallel repetition)**.** *Let G'_2 be a (ε, δ)-fortified game against classical substrategies. Then for any game G'_1 we have*

$$\textsc{Val}(G'_1 \otimes G'_2) \leq (\textsc{Val}(G'_2) + \varepsilon) \cdot \textsc{Val}(G'_1) + \delta \cdot |\Sigma_{G'_1}|, \tag{32}$$

where $\Sigma_{G'_1}$ is the total answer alphabet size (i.e. the product of Alice and Bob's alphabets) of G'_1.

[10] The term $\mu(S \times T) = \mathbb{E}_{(x,y)\sim\mu} f(x)g(y)$ is a natural scaling parameter playing an important role in our discussion as a measure of the "largeness" of a subgame or a substrategy.

We prove this theorem in Section 4 by adapting the proof of the analogous theorems in [28, 6] to the analytic setting. Unfortunately, while this theorem exemplifies the main idea behind our results, it is not directly useful for applications. The reason for this is that the fortification procedure $G \to G'$ via concatenation induces a large blow-up in the alphabet size, $|\Sigma_{G'}| \approx |\Sigma_G|^D$, where $D = \frac{1}{\varepsilon\sqrt{\delta}}$ is the degree of the expander graph chosen. As one iterates the repetition procedure m times, the blow-up due to the additive term in (32) will be of order $\delta|\Sigma_{G'}|^{m-1}$. But typically $|\Sigma_{G'}|^{m-1} \gg |\Sigma_G|^{(m-1)/\sqrt{\delta}}$, leading to a term larger than 1 and rendering the theorem useless.

We resolve this problem by proving an improved repetition theorem which exploits the fact that G' takes the form of a concatenated game, whose inner game G has a much smaller alphabet.

▶ **Theorem 22.** *Let G' be a concatenated game, with inner game G, that is (ε, δ)-weakly fortified against classical substrategies. If $\delta \cdot (m-1) \cdot |\Sigma_G|^{m-1} \leq \eta$ then*

$$\textsc{Val}(G'^{\otimes m}) \leq (\textsc{Val}(G) + \varepsilon)^m + \eta. \tag{33}$$

Similarly, if G' is (ε, δ) weakly-fortified against entangled substrategies and $\delta \cdot (m-1) \cdot |\Sigma_G|^{m-1} \leq \eta$ then

$$\textsc{Val}^*(G'^{\otimes m}) \leq (\textsc{Val}^*(G) + \varepsilon)^m + \eta. \tag{34}$$

The main advantage of Theorem 22 compared to Theorem 21 is in the additive error, which is now is in terms $|\Sigma_G|$ rather than $|\Sigma_{G'}|$. What is important here is that the size of $|\Sigma_G|$ is independent of the fortification parameters (ε, δ) whereas $|\Sigma_{G'}|$ grows exponentially as δ decreases. Let us also note that Theorem 22 is quite general, and in particular applies to the multiplayer case.

3.3 Gap amplification

Having stated our main parallel repetition, fortification, and biregularization theorems, all the main components of gap amplification are finally in place. Indeed, using $\textsc{Val}(G) = \textsc{Val}(G')$ Theorem 22 implies our final gap amplification for the classical value. This matches the parameters of main results of [28, 6] and extends it to more general settings.

Since quantumly we could have $\textsc{Val}^*(G') < \textsc{Val}^*(G)$, from (34) we cannot obtain

$$\textsc{Val}^*(G'^{\otimes m}) \leq (\textsc{Val}^*(G') + \varepsilon)^m + \eta. \tag{35}$$

However, Theorem 3 and Theorem 22 are still sufficient to prove a gap amplification theorem for the case where the completeness holds against classical players and the soundness against the quantum ones.[11] To obtain a fully quantum gap amplification however, we need to appeal to the notion of *ordered fortification* which, as we discussed, is a entangled-value preserving variant of the ordinary fortification.

▶ **Theorem 23** (Theorem 5 restated)**.** *Let G be a game and M and P be two bipartite graphs as above. Let G'_{OF} be constructed from G and $G' = (M \circ G \circ P)$ as in Definition 4. Then, we have*

$$\textsc{Val}^*(G'_{OF}) = \textsc{Val}^*(G).$$

Furthermore if M and P are λ-spectral expanders and $\lambda \leq \frac{\varepsilon^2\delta}{56}$, then G'_{OF} is also (ε, δ) weakly fortified.

[11] E.g. as was the case in [23, 32].

We stress that G'_{OF} constructed above is itself a concatenated game with the inner game $G^{\oplus l}$, disjoint union of $l = \mathrm{poly}(\frac{1}{\varepsilon^2\delta})$ copies of G. This means the inner alphabet size of G'_{OF} is precisely the same as G's, and therefore there is fortunately no issue in terms of alphabet blow-up for applying Theorem 22 to G'_{OF}. So using G'_{OF} instead of G' in Theorem 22, we can finally prove the analogue of (35) for G'_{OF}.

Parameters of gap amplification

We can now discuss the parameters of the gap amplification corollaries. As in [28, 6], the parameters are typically very good in terms of question sizes but much worse in terms of alphabet size. Here, we mostly focus our discussion to gap amplification in the classical setting as the calculations in the quantum setting are similar.

To understand the parameters, we need to only consider the soundness case. Suppose we are given a game G with guarantee $\textsc{Val}(G) \le 1-\tau$ and a target soundness value β. We choose $\varepsilon = \tau/2$ and m such that $(\textsc{Val}(G)+\varepsilon)^m \le \beta/2$. Hence, we have $m = \frac{\log(2/\beta)}{\log(1-\tau/2)} \le \frac{2\log(2/\beta)}{\tau}$. We want

$$\textsc{Val}(G'^{\otimes m}) \le (\textsc{Val}(G)+\varepsilon)^m + \delta\cdot(m-1)|\Sigma_G|^{m-1} \le \beta. \tag{36}$$

Hence, we just need to ensure $\delta\cdot|\Sigma_G|^{m-1} \le \beta/2$. So we have $\delta = \frac{\beta}{(m-1)\cdot|\Sigma_G|^{m-1}}$.

So what does the above mean in terms of the size of the final output of gap amplification $G'^{\otimes m}$. The question size is $|X|^m$ and $|Y|^m$ (since we have $|X'| = |X|$ and $|Y'| = |Y|$). Note that m is essentially as small as we can hope for because even given a perfect parallel repetition theorem, we had to take $m \approx \frac{\log(1/\beta)}{\tau}$. Hence, the construction is essentially optimal in terms of question sizes.

For the alphabet size, the situation is much worse. We have $|\Sigma_{G'}| = |\Sigma_G|^D$ where $D = O(\frac{\mathrm{poly}\log(1/\varepsilon^2\delta)}{\varepsilon^2\delta})$. This means (up to dominant factors) that $|\Sigma_{G'^{\otimes m}}| = |\Sigma_G|^{\frac{m^2|\Sigma_G|^{m-1}}{\beta}}$ which means that the alphabet is exponentially worse than basic parallel repetition which results in $|\Sigma_G|^m$. Note that however in typical settings where $|\Sigma_G|$ is constant and β a small constant (or inverse logarithmic in size of G), this exponentially worse behavior of alphabet size does not cause a significant problem.

Next, let us consider the setting where the completeness holds for classical players and soundness against entangled players. In this case, we can just use Theorem 3 instead of Theorem 2, and hence all the calculations are precisely the same with ε and δ replaced with their squares.

Lastly, in the fully quantum case we need to use Theorem 5. In this case, m, ε, δ are chosen in precisely the same way. Alphabet size is also exactly the same as G'_{OF} has the same alphabet size as G'. The only difference is that the question sizes in G'_{OF} are slightly larger than G': we have $|X'| = |X|\cdot\mathrm{poly}(\frac{|\Sigma_G|^m}{\beta})$ and $|Y'| = |Y|\cdot\mathrm{poly}(\frac{|\Sigma_G|^m}{\beta})$. This is however arguably a minor blow-up since we typically expect that $|\Sigma_G|/\beta$ to be much smaller than $\mathit{size}(G) = |X|\cdot|Y|$.

4 Parallel Repetition Theorems

In this section we prove our main parallel repetition theorem.

▶ **Theorem 24** (Theorem 22 restated). *Let G' be a concatenated game (ε,δ)-weakly fortified against classical substrategies with inner game G. If $\delta\cdot(m-1)\cdot|\Sigma_G|^{m-1} \le \eta$ then*

$$\textsc{Val}(G'^{\otimes m}) \le (\textsc{Val}(G)+\varepsilon)^m + \eta. \tag{37}$$

Similarly, if G' is (ε, δ) weakly-fortified against entangled substrategies and $\delta \cdot (m-1) \cdot |\Sigma_G|^{m-1} \leq \eta$ then

$$\textsc{Val}^*(G'^{\otimes m}) \leq (\textsc{Val}^*(G) + \varepsilon)^m + \eta. \tag{38}$$

The proof follows directly from the following proposition.

▶ **Proposition 25.** *Let $\{G'_i\}_{i=1}^t$ be a collection of concatenated games with inner games $\{G_i\}_{i=1}^t$. Suppose that G'_t is (ε, δ) weakly fortified against classical substrategies. Then,*

$$\textsc{Val}(G'_1 \otimes G'_2 \otimes \ldots \otimes G'_t) \leq (\textsc{Val}(G_t) + \varepsilon) \cdot \textsc{Val}(G'_1 \otimes G'_2 \otimes \ldots \otimes G'_{t-1}) + \delta \cdot \prod_{i=1}^{t-1} |\Sigma_{G_i}|. \tag{39}$$

Similarly, if G'_t is (ε, δ) weakly fortified against quantum substrategies, then

$$\textsc{Val}^*(G'_1 \otimes G'_2 \otimes \ldots \otimes G'_t) \leq (\textsc{Val}^*(G_t) + \varepsilon) \cdot \textsc{Val}^*(G'_1 \otimes G'_2 \otimes \ldots \otimes G'_{t-1}) + \delta \cdot \prod_{i=1}^{t-1} |\Sigma_{G_i}|. \tag{40}$$

The key to proving Proposition 25 is to work with the induced strategies. This allows us to get an additive error depending just on the alphabet size of the inner game. In the proof, we use the usual notation where a strategy missing an (answer) argument indicates summation over that variable. For example,

$$f(x_1, a_1, \ldots, x_{t-1}, a_{t-1}, x_t) = \sum_{a_t} f(x_1, a_1, \ldots, x_{t-1}, a_{t-1}, x_t, a_t).$$

Proof. We only prove (39) as the proof of (40) follows the same structure. Also for simplicity, we focus on the case of two-player games as the proof of the multiplayer case is a straightforward extension.

Consider any strategies $f : X'_1 \times A'_1 \times \ldots \times X'_t \times A'_t \to [0,1]$, $g : Y'_1 \times B'_1 \times \ldots \times Y'_t \times B'_t \to [0,1]$. To clarify notation we will denote tuples $(z_1, \ldots, z_{t-1})$ as $\mathbf{z}_{<t}$. With this notation, $\textsc{Val}(G'_1 \otimes \ldots \otimes G'_t, f, g)$ is precisely

$$\mathop{\mathbb{E}}_{(\mathbf{x}_{\leq t}, \mathbf{y}_{\leq t})} \mathop{\mathbb{E}}_{\mathbf{x'}_{\leq t}} \mathop{\mathbb{E}}_{\mathbf{y'}_{\leq t}} \sum_{\mathbf{a'}_{\leq t}, \mathbf{b'}_{\leq t}} \prod_{i=1}^{t} V(a'_i(x_i), b'_i(y_i), x_i, y_i)\, f(\mathbf{x'}_{\leq t}, \mathbf{a'}_{\leq t}) \cdot g(\mathbf{y'}_{\leq t}, \mathbf{b'}_{\leq t}), \tag{41}$$

where the expectations are according to $(x_i, y_i) \sim \mu_i$ and $x'_i \sim N(x_i)$ and $y'_i \sim N(y_i)$ for all $i = 1, \ldots, t$. As usual let

$$f(\mathbf{x}_{<t}, \mathbf{a}_{<t}, x'_t, a'_t) = \mathop{\mathbb{E}}_{\mathbf{x'}_{<t} \sim N(\mathbf{x}_{<t})} \sum_{a'_i(x_i) = a_i,\, i<t} f(\mathbf{x'}_{<t}, \mathbf{a'}_{<t}, x'_t, a'_t). \tag{42}$$

Using this notation, we can rewrite (41) as

$$\mathop{\mathbb{E}}_{(\mathbf{x}_{<t}, \mathbf{y}_{<t})} \sum_{\mathbf{a}_{<t}, \mathbf{b}_{<t}} \prod_{i=1}^{t-1} V(a_i, b_i, x_i, y_i)\, S(\mathbf{x}_{<t}, \mathbf{y}_{<t}, \mathbf{a}_{<t}, \mathbf{b}_{<t}), \tag{43}$$

where $S(\mathbf{x}_{<t}, \mathbf{y}_{<t}, \mathbf{a}_{<t}, \mathbf{b}_{<t})$ is given by

$$\mathop{\mathbb{E}}_{(x_t, y_t)} \mathop{\mathbb{E}}_{x'_t} \mathop{\mathbb{E}}_{y'_t} \sum_{a'_t, b'_t} V(a'_t(x_t), b'_t(y_t), x_t, y_t)\, f(\mathbf{x}_{<t}, \mathbf{a}_{<t}, x'_t, a'_t) \cdot g(\mathbf{y}_{<t}, \mathbf{b}_{<t}, y'_t, b'_t). \tag{44}$$

Consider the following substrategy G'_t: fix the first $2(t-1)$ arguments of f to $(\mathbf{x}_{<t}, \mathbf{a}_{<t})$ and the first $2(t-1)$ arguments of g to $(\mathbf{y}_{<t}, \mathbf{b}_{<t})$. Then (44) is precisely the value of this substrategy in G'_t. Since G'_t is (ε, δ) weakly fortified, it follows that

$$(44) \leq (\textsc{Val}(G_t) + \varepsilon) \cdot \mathop{\mathbb{E}}_{(x_t, y_t)} f(\mathbf{x}_{<t}, \mathbf{a}_{<t}, x_t) \cdot g(\mathbf{y}_{<t}, \mathbf{b}_{<t}, y_t) + \delta. \tag{45}$$

Plugging this expression back into (43), $\textsc{Val}(G'_1 \otimes \ldots \otimes G'_t, f, g)$ is bounded by

$$(\textsc{Val}(G_t) + \varepsilon) \mathop{\mathbb{E}}_{(\mathbf{x}_{\leq t}, \mathbf{y}_{\leq t})} \sum_{\mathbf{a}_{<t}, \mathbf{b}_{<t}} \prod_{i=1}^{t-1} V(a_i, b_i, x_i, y_i) f(\mathbf{x}_{<t}, \mathbf{a}_{<t}, x_t) g(\mathbf{y}_{<t}, \mathbf{b}_{<t}, y_t) + \delta \prod_{i=1}^{t-1} |\Sigma_{G_i}|.$$

To conclude we observe that

$$\mathop{\mathbb{E}}_{(\mathbf{x}_{\leq t}, \mathbf{y}_{\leq t})} \sum_{\mathbf{a}_{<t}, \mathbf{b}_{<t}} \prod_{i=1}^{t-1} V(a_i, b_i, x_i, y_i)\, f(\mathbf{x}_{<t}, \mathbf{a}_{<t}, x_t) \cdot g(\mathbf{y}_{<t}, \mathbf{b}_{<t}, y_t) \tag{46}$$

is at most $\textsc{Val}(G'_1 \otimes \ldots \otimes G'_{t-1})$, as for any fixed (x_t, y_t) the functions $f(\cdot, x_t) : X'_1 \times A'_1 \times \ldots \times X'_{t-1} \times A'_{t-1} \to [0,1]$ and $g(\cdot, y_t) : Y'_1 \times B'_1 \times \ldots \times Y'_{t-1} \times B'_{t-1} \to [0,1]$ are valid strategies in $G'_1 \otimes \ldots \otimes G'_{t-1}$. ◀

▶ Remark. Theorem 21 immediately follows from Proposition 25 by taking $t = 2$ and considering the trivial concatenation $G'_1 = G_1$, $G'_2 = G_2$.

Theorem 22 follows easily.

Proof of Theorem 22. We prove (37) as the proof of (38) is similar.

The proof is by induction on m. The case $m = 1$ is clear. By the induction hypothesis we have

$$\textsc{Val}(G'^{\otimes(m-1)}) \leq (\textsc{Val}(G) + \varepsilon)^{m-1} + \delta \cdot (m-2)|\Sigma_G|^{m-2}.$$

Note that we can assume $\textsc{Val}(G) + \varepsilon < 1$ otherwise (37) holds trivially. Applying Proposition 25 we see that

$$\begin{aligned}
\textsc{Val}(G'^{\otimes m}) &\leq (\textsc{Val}(G) + \varepsilon) \cdot \textsc{Val}(G'^{\otimes(m-1)}) + \delta \cdot |\Sigma_G|^{m-1} \\
&\leq (\textsc{Val}(G) + \varepsilon)^m + \delta \cdot (\textsc{Val}(G) + \varepsilon) \cdot (m-2)|\Sigma_G|^{m-2} + \delta \cdot |\Sigma_G|^{m-1} \\
&\leq (\textsc{Val}(G) + \varepsilon)^m + \delta \cdot (m-1) \cdot |\Sigma_G|^{m-1}.
\end{aligned}$$ ◀

5 Classical Fortification

In this section we prove our main theorem regarding the fortification of classical games. Beside providing a short and self-contained treatment of the main result of [28, 6], it serves as preparation for the analysis of Section 7.

▶ **Theorem 26** (Theorem 2 restated). *Let G be a biregular game, M and P two bipartite λ-spectral expanders. If $\lambda \leq \frac{\varepsilon}{2}\sqrt{\frac{\delta}{2}}$, then the concatenated game $G' = (M \circ G \circ P)$ is (ε, δ) weakly fortified against classical substrategies.*

We note that it follows from [6, Appendix C] that the dependence λ and δ in Theorem 2 is up to constant factors optimal. On the other hand, the tightness of dependence of ε and δ does not seem to follow from [6] lower bound (however, δ is by far the more significant of the two parameters).

5.1 Proof of Theorem 2

We start with a simple claim whose proof we will defer to the end of the subsection.

▶ **Claim 27.** *Let $M = (X' \times X, E)$ and $N = (Y' \times Y, F)$ be two biregular bipartite graphs that are λ-spectral expanders. Let μ be a distribution on $X \times Y$ such that both marginals of μ are uniform. Let $f : X' \to \mathbb{R}$ and $g : Y' \to R$ be any functions, and denote $f : X \to \mathbb{R}$ and $g : Y \to \mathbb{R}$ the functions $f : x \mapsto \mathbb{E}_{x' \sim N(x)} f(x)$, $g : y \mapsto \mathbb{E}_{y' \sim N(y)} g(y)$ respectively. Then*

$$\mathop{\mathbb{E}}_{(x_1,y_1)\sim\mu} \Big| f(x_1)g(y_1) - \mathop{\mathbb{E}}_{(x_2,y_2)\sim\mu} f(x_2)g(y_2) \Big| \le 2\sqrt{2}\lambda \Big(\mathop{\mathbb{E}}_{x'\sim X'} |f(x')|^2 \Big)^{1/2} \Big(\mathop{\mathbb{E}}_{y'\sim Y'} |g(y')|^2 \Big)^{1/2}$$

and

$$\Big| \mathop{\mathbb{E}}_{x_1\sim X} f(x) \mathop{\mathbb{E}}_{y_1\sim Y} g(y_1) - \mathop{\mathbb{E}}_{(x_2,y_2)\sim\mu} f(x_2)g(y_2) \Big| \le 2\lambda^2 \Big(\mathop{\mathbb{E}}_{x'\sim X'} |f(x')|^2 \Big)^{1/2} \Big(\mathop{\mathbb{E}}_{y'\sim Y'} |g(y')|^2 \Big)^{1/2}.$$

We prove a slightly stronger statement which implies Theorem 2. Let f, g be any substrategies for G, and let $\gamma = \mathbb{E}_{(x,y)\sim\mu} f(x)g(y)$. We claim that

$$\textsc{Val}(G', f, g) \le \textsc{Val}(G)\gamma + 2\sqrt{2}\,\lambda\sqrt{\gamma} + 4\,\lambda^2. \tag{47}$$

To deduce the bound claimed in Theorem 2 from (47) we distinguish two cases. Either $\gamma \le \delta$, in which case using the trivial estimate $\textsc{Val}(G', f, g) \le \gamma$ the bound immediately follows. Or $\gamma > \delta$, in which case

$$\begin{aligned} \textsc{Val}(G)\gamma + 2\sqrt{2}\lambda\sqrt{\gamma} + 4\lambda^2 &\le \gamma(\textsc{Val}(G) + 2\sqrt{2}\lambda\delta^{-1/2}) + 4\lambda^2 \\ &\le \gamma(\textsc{Val}(G) + \varepsilon) + \delta \end{aligned}$$

given the relation between ε, δ and λ expressed in the theorem.

It remains to prove (47). Fix substrategies f and g. We have

$$\begin{aligned} \textsc{Val}(G', f, g) &= \mathop{\mathbb{E}}_{(x,y)\sim\mu} \sum_{V(a,b,x,y)=1} f(x,a) \cdot g(y,b) \\ &= \mathop{\mathbb{E}}_{(x,y)\sim\mu} f(x)g(y) \sum_{V(a,b,x,y)=1} \frac{f(x,a)}{f(x)} \cdot \frac{g(y,b)}{g(y)}, \end{aligned}$$

where we adopt the convention that $0/0 = 0$. Using the triangle inequality,

$$\begin{aligned} \textsc{Val}(G', f, g) &\le \gamma \mathop{\mathbb{E}}_{(x,y)\sim\mu} \sum_{V(a,b,x,y)=1} \frac{f(x,a)}{f(x)} \cdot \frac{g(y,b)}{g(y)} + \mathop{\mathbb{E}}_{(x,y)\sim\mu} \big| f(x)g(y) - \gamma \big| \\ &\le \gamma\, \textsc{Val}(G) + \mathop{\mathbb{E}}_{(x,y)\sim\mu} \big| f(x)g(y) - \gamma \big|, \end{aligned} \tag{48}$$

where the second inequality follows since $(x, a) \mapsto f(x, a)/f(x)$ and $(y, b) \mapsto g(y, b)/g(y)$ form a valid pair of strategies for G. It remains to estimate the second term above. Applying the first inequality in Claim 27,

$$\begin{aligned} \mathop{\mathbb{E}}_{(x,y)\sim\mu} \big| f(x)g(y) - \gamma \big| &\le 2\sqrt{2}\lambda \Big(\mathop{\mathbb{E}}_{x'\sim X'} |f(x')|^2 \Big)^{1/2} \Big(\mathop{\mathbb{E}}_{y'\sim Y'} |g(y')|^2 \Big)^{1/2} \\ &\le 2\sqrt{2}\lambda \Big(\mathop{\mathbb{E}}_{x'\sim X'} f(x') \mathop{\mathbb{E}}_{y'\sim Y'} g(y') \Big)^{1/2} \\ &\le 2\sqrt{2}\lambda \sqrt{\gamma + 2\lambda^2} \\ &\le 2\sqrt{2}\lambda (\sqrt{\gamma} + \sqrt{2}\lambda) \\ &= 2\sqrt{2}\lambda\sqrt{\gamma} + 4\lambda^2, \end{aligned}$$

where in the second inequality we used $0 \le f(x'), g(y') \le 1$ for all x', y' and the third uses the second inequality in Claim 27. Together with (48) this proves (47).

Finally, we prove Claim 27.

Proof of Claim 27. For the first inequality, write

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(x_1,y_1)\sim\mu}\big|f(x_1)g(y_1) - \mathop{\mathbb{E}}_{(x_2,y_2)\sim\mu} f(x_2)g(y_2)\big| \\
&\qquad\le \mathop{\mathbb{E}}_{(x_1,y_1),(x_2,y_2)\sim\mu}\big(|f(x_1)-f(x_2)||g(y_1)| + |f(x_2)||g(y_1)-g(y_2)|\big) \\
&\qquad\le \Big(\mathop{\mathbb{E}}_{x_1,x_2\sim X}|f(x_1)-f(x_2)|^2\Big)^{1/2}\Big(\mathop{\mathbb{E}}_{y_1\sim Y}|g(y_1)|^2\Big)^{1/2} \\
&\qquad\qquad\qquad + \Big(\mathop{\mathbb{E}}_{x_2\sim X}|f(x_2)|^2\Big)^{1/2}\Big(\mathop{\mathbb{E}}_{y_1,y_2\sim Y}|g(y_1)-g(y_2)|^2\Big)^{1/2} \\
&\qquad\le \lambda\Big(\mathop{\mathbb{E}}_{x_1',x_2'\sim X'}|f(x_1')-f(x_2')|^2\Big)^{1/2}\Big(\mathop{\mathbb{E}}_{y_1'\sim Y'}|g(y_1')|^2\Big)^{1/2} \\
&\qquad\qquad\qquad + \lambda\Big(\mathop{\mathbb{E}}_{x_2'\sim X'}|f(x_2')|^2\Big)^{1/2}\Big(\mathop{\mathbb{E}}_{y_1',y_2'\sim Y'}|g(y_1')-g(y_2')|^2\Big)^{1/2},
\end{aligned}$$

where the last inequality uses Proposition 18. Now note that $\mathbb{E}_{x_1',x_2'\sim X'}|f(x_1')-f(x_2')|^2 \le 2\,\mathbb{E}_{x'\sim X'}|f(x')|^2$. Applying a similar bound for g gives us the first inequality. For the second, write

$$\begin{aligned}
&\Big|\mathop{\mathbb{E}}_{(x_2,y_2)\sim\mu} f(x_2)g(y_2)\mathop{\mathbb{E}}_{x_1\sim X} f(x)\mathop{\mathbb{E}}_{y_1\sim Y} g(y_1)\Big| \\
&\qquad= \Big|\mathop{\mathbb{E}}_{(x_2,y_2)\sim\mu,x_1\sim X,y_1\sim Y}(f(x_1)-f(x_2))(g(y_1)-g(y_2))\Big| \\
&\qquad\le \Big(\mathop{\mathbb{E}}_{x_1,x_2\sim X}(f(x_1)-f(x_2))^2\Big)^{1/2}\Big(\mathop{\mathbb{E}}_{y_1,y_2\sim Y}(g(y_1)-g(y_2))^2\Big)^{1/2} \\
&\qquad\le \lambda^2\Big(\mathop{\mathbb{E}}_{x_1',x_2'\sim X'}(f(x_1)-f(x_2))^2\Big)^{1/2}\Big(\mathop{\mathbb{E}}_{y_1',y_2'\sim Y'}(g(y_1')-g(y_2'))^2\Big)^{1/2} \\
&\qquad\le 2\lambda^2\Big(\mathop{\mathbb{E}}_{x'\sim X'}|f(x')|^2\Big)^{1/2}\Big(\mathop{\mathbb{E}}_{y'\sim Y'}|g(y')|^2\Big)^{1/2}.
\end{aligned}$$

◀

5.2 A simple multiplayer fortification

The following is a simple fortification theorem for k-player games. Since Theorem 22 applies equally well to the multiplayer setting, we get a hardness amplification result for classical multiplayer games.

▶ **Theorem 28.** *Let G be a k-player game. Suppose G' is given by composing each of the k sides of G by a λ-spectral expander where $\lambda \le 2\delta/k$. Then G' is a $(0,\delta)$ fortified game.*[12]

Proof. Consider a classical substrategy for G' given by $f_i : X_i' \times A_i' \to \mathbb{R}^+$ for $i = 1, 2, \ldots, k$. As usual, denote $f_i : X_i \times A_i \to \mathbb{R}^+$ the projection of f_i to the inner game G. By definition,

$$\textsc{Val}(G, \{f_i\}_{i=1}^k) = \mathop{\mathbb{E}}_{(x_1,\ldots,x_k)} \sum_{a_1,a_2,\ldots,a_k} V(a_1,\ldots,a_k,x_1,\ldots,x_k)\prod_{j=1}^k f_j(x_j,a_j).$$

[12] Although there is no ε dependence in the above, when applied to 2-player games the theorem is still weaker than Theorem 2 because of the worse dependence on δ – which is the more crucial parameter than ε.

We can rewrite the above as

$$\mathop{\mathbb{E}}_{(x_1,\ldots,x_k)} \prod_{i=1}^{k} f_i(x_i) \sum_{a_1,\ldots,a_k} V(a_1,\ldots,a_k,x_1,\ldots,x_k) \cdot \frac{f_1(x_1,a_1)\cdot f_x(x_2,a_2)\ldots f_k(x_k,a_k)}{f(x_1)\cdot f(x_2)\cdot\ldots\cdot f(x_k)}.$$

Let $\gamma = \mathbb{E}_{(x_1,\ldots,x_k)} \prod_{i=1}^{k} f_i(x_i)$. Applying the triangle inequality,

$$\textsc{Val}(G,\{f_i\}_{i=1}^k) \leq \gamma \cdot \textsc{Val}(G) + \mathop{\mathbb{E}}_{(x_1,\ldots,x_k)} |\prod_{i=1}^{k} f_i(x_i) - \gamma|.$$

To conclude it will suffice to show the second term above is at most δ. Let $\overline{f_i} = \mathbb{E}_{x_i} f(x_i)$. Then

$$\begin{aligned}
\mathop{\mathbb{E}}_{(x_1,\ldots,x_k)} \left|\prod_{i=1}^{k} f_i(x_i) - \gamma\right| &\leq \mathop{\mathbb{E}}_{x_1,\ldots,x_k} \left|\prod_{i=1}^{k} f_i(x_i) - \prod_{i=1}^{k} \overline{f_i}\right| + \mathop{\mathbb{E}}_{(x_1,\ldots,x_k)} \left|\prod_{i=1}^{k} \overline{f_i} - \gamma\right| \\
&= \mathop{\mathbb{E}}_{(x_1,\ldots,x_k)} \left|\prod_{i=1}^{k} f_i(x_i) - \prod_{i=1}^{k} \overline{f_i}\right| + \left|\prod_{i=1}^{k} \overline{f_i} - \mathop{\mathbb{E}}_{x_1,\ldots,x_k} \prod_{i=1}^{k} f_i(x_i)\right| \\
&\leq 2 \cdot \mathop{\mathbb{E}}_{(x_1,\ldots,x_k)} \left|\prod_{i=1}^{k} f_i(x_i) - \prod_{i=1}^{k} \overline{f_i}\right| \\
&\leq 2 \sum_{i=1}^{k} \mathop{\mathbb{E}}_{x_i} |f_i(x_i) - \overline{f_i}|,
\end{aligned}$$

where the first equality is by definition of γ, the second inequality by convexity of $|\cdot|$, and the last follows from

$$\begin{aligned}
&|f_1(x_1)f_2(x_2)\ldots f_k(x_k) - \overline{f_1 f_2}\ldots\overline{f_k}| \\
&\leq \sum_{\ell=1}^{k} \left|\prod_{i=1}^{\ell-1} f_i(x_i)\cdot \prod_{i=\ell}^{k} \overline{f_i} - \prod_{i=1}^{\ell} f_i(x_i)\cdot \prod_{i=\ell+1}^{k} \overline{f_i}\right| \\
&\leq \sum_{i=1}^{k} \mathop{\mathbb{E}}_{x_i} |f_i(x_i) - \overline{f_i}|.
\end{aligned}$$

Hence,

$$\mathop{\mathbb{E}}_{x_1,\ldots,x_k} \left|\prod_{i=1}^{k} f_i(x_i) - \gamma\right| \leq 2\sum_{i=1}^{k} \left(\mathop{\mathbb{E}}_{x_i}(f_i(x_i) - \overline{f_i})^2\right)^{1/2} \leq 2\lambda \sum_{i=1}^{k} \left(\mathop{\mathbb{E}}_{x_i'}(f_i(x_i') - \overline{f_i})^2\right)^{1/2} \leq 2\lambda k.$$

The desired result follows. ◀

6 Reducing Strong to Weak Fortification for Entangled Games

In this section, we start working toward the problem of fortifying games in the entangled case. In particular, we show how Theorem 5 follows from Theorem 3. Let $G = (X \times Y, A \times B, \mu, V)$ be a two-player game.

▶ **Definition 29.** For a game G and integer $l \in \mathbb{N}$ let $G^{\oplus l}$ denote the disjoint union of l copies of G.

Suppose that M and P are regular bipartite graphs over $X' \times X$ and $Y' \times Y$, respectively. Suppose further that M and P are balanced, i.e. $|X'| = |X|$ and $|Y'| = |Y|$. Let d_M and d_N denote the degree of vertices M and P, respectively. (Note that since the graphs are balanced and regular, the left and right degrees are the same.)

Following [6], we assume that M and P are explicit bipartite almost-Ramanujan expanders, as provided e.g. by [7], for which the second-largest singular values λ_M and λ_P of A_M and A_P (the normalized adjacency matrices) respectively satisfy

$$\lambda_M = O\left(\frac{\mathrm{poly}(\log d_M)}{\sqrt{d_M}}\right), \qquad \lambda_P = O\left(\frac{\mathrm{poly}(\log d_P)}{\sqrt{d_P}}\right). \tag{49}$$

Note that if $d_M, d_P = \widetilde{\Omega}(\frac{1}{\varepsilon^2\delta})$ then Theorem 3 implies that $G' = (M \circ G \circ P)$ is (ε, δ) weakly-fortified. Next we recall the definition of G'_{OF} from the introduction.

Ordered fortification

Let G, M, P and $G' = (M \circ G \circ P)$ be as above. let $l = \max\{d_M, d_P\}$. In G'_{OF_l} (or simply G'_{OF} where $l = \max\{d_M, d_P\}$) the referee samples questions (x, y) as in G and selects two random neighbors $x' \in X'$ and $y' \in Y'$ of x and y in M and P respectively. Then the referee selects two random injective maps $r_{x'} : N(x') \to [l]$ and $s_{y'} : N(y') \to [l]$ under the condition $r_{x'}(x) = s_{y'}(y)$. Alice's question then is the pair $(x', r_{x'})$ and Bob's is the pair $(y', s_{y'})$. Alice outputs an answer tuple $a' : N(x') \to A$ and Bob $b' : N(y') \to B$. The players win if $V(a'(x), b'(y), x, y) = 1$.

▶ Remark. Note that G'_{OF} has exactly the same answer alphabet size as G', the question sizes $|X'_{OF}|$ and $|Y'_{OF}|$ are larger than in G'. This blow-up can be mitigated as follows. It turns out that in Definition 30 the use of the complete set $S_{(d,l)}$ is unnecessary. More precisely, from the proof of the main claim of this section, Claim 33 below, it will be clear that the only condition required is that the permutations be chosen from a pairwise independent subset of $S_{(d,l)}$. Selecting the smallest possible such subset lets us reduce the blow-up in the size of the question sets from a multiplicative $D!$ down to $\mathrm{poly}(D) = \mathrm{poly}(\frac{1}{\varepsilon^2\delta})$. We omit the details.

Although it may not be immediately apparent, it is possible to view G'_{OF} as a concatenated game. Let $G^{\oplus l}$ be as in Definition 29. Note that $G^{\oplus l}$ has exactly the same classical and entangled value as G. Let $S_{(d_M,l)}$ denote the set of all injective maps from $[d_M] \to [l]$. Fix maps $u_{x'} : N(x') \to [d_M]$ and $v_{y'} : N(y') \to [d_M]$ ordering the neighborhoods of each x', y' in an arbitrary way.

▶ **Definition 30.** Let M be a regular bipartite graph over $X' \times X$ as above . We define $\widetilde{M}$ as a bipartite graph over $X'_{OF} := X' \times S_{(d_M,l)}$ and $X_{OF} := X \times [l]$ where

$$(x', \pi) \sim_{\widetilde{M}} (x, i) \qquad \Longleftrightarrow \qquad \pi(u_{x'}(x)) = i.$$

We define $\widetilde{P}$ from P in a similar way.

Note that here $\pi \circ u_{x'}$ exactly corresponds to $r_{x'} : N(x') \to [l]$ map from the original definition of G'_{OF}. Hence, we obtain the following alternative characterization of G'_{OF}.

▶ **Proposition 31.** *The game G'_{OF} constructed above is a concatenated game given by*

$$G'_{OF} = (\widetilde{M} \circ G^{\oplus l} \circ \widetilde{P}).$$

Next, we show that ordered fortification preserves the entangled value (the classical value is also preserved but that is not important here).

▶ **Proposition 32.** *We have* $\textsc{Val}^*(G'_{OF}) = \textsc{Val}^*(G)$.

Proof. In one direction we have $\textsc{Val}^*(G'_{OF}) \leq \textsc{Val}^*(G^{\oplus l}) = \textsc{Val}^*(G)$ where we used Propositions 12 and 31. For the other direction, consider any entangled strategy $(|\psi\rangle, \{A_x^a\}, \{B_y^b\})$ for G. We construct a strategy for $G'_{\oplus}$ that achieves the same value. The provers share l copies of the state $|\psi\rangle$, and each copy is assigned a unique label $i \in [l]$. Alice and Bob receive questions $(x', r_{x'})$ and $(y', s_{y'})$, respectively. For each $x \in N(x')$, Alice applies $\{A_x^a\}$ to the $r_{x'}(x)$-th copy of $|\psi\rangle$. Bob applies a similar strategy.

Since by construction the "true questions" x^* and y^* are given the same label, the distribution of answers obtained for x^* and y^* is identical to the distribution of answers obtained while playing G using $(|\psi\rangle, \{A_x^a\}, \{B_y^b\})$, hence achieving the same winning probability. ◀

The main technical step in reducing Theorem 5 to Theorem 3 is an analysis of the singular values of $\widetilde{M}, \widetilde{N}$ in terms of the singular values of M and N. We prove the following.

▶ **Claim 33.** *Let M be a bipartite graph over $X' \times X$ as above and let λ_M denote the second largest singular value of M. Let $\tilde{M}$ be as in Definition 30. Then,*

$$\lambda_{\tilde{M}} \leq \max\left\{\lambda_M, \frac{1}{\sqrt{d_M - 1}}\right\}.$$

Since in our case $\lambda_M = O\left(\mathrm{poly}(\log d_M)/d_M\right)$, Claim 33 implies that $\lambda_{\tilde{M}}$ satisfies the same bound. Also note that a similar statement of course applies to $\tilde{P}$ and $\lambda_{\tilde{P}}$. So we see that Theorem 3, Propositions 32 and 31, and Claim 33 together imply Theorem 5; it remains to prove the latter.

Proof of Claim 33. Recall that by assumption M is a regular balanced bipartite graph. Let $d := d_M$ the degree of vertices in M. The normalized adjacency matrix of $\tilde{M}$ is given by

$$A_{\tilde{M}}((x', \pi), (x, i)) = \begin{cases} \frac{1}{d} \cdot \sqrt{\frac{(l-d)!}{(l-1)!}} & (x', \pi) \sim_{\tilde{M}} (x, i) \\ 0 & (x', \pi) \not\sim_{\tilde{M}} (x, i) \end{cases}. \tag{50}$$

We relate the second largest singular value λ_M of A_M and the second largest singular value $\lambda_{\tilde{M}}$ of $\tilde{M}$ by relating the eigenvalues of $B = A_M^\top A_M$ and $C = A_{\tilde{M}}^\top A_{\tilde{M}}$. We can explicitly compute the entries of B and C. For B,

$$B(x_1, x_2) = \frac{|\{x' \in X' : \{x_1, x_2\} \subset N(x')\}|}{d^2}, \tag{51}$$

and in particular $B(x, x) = \frac{1}{d}$ for all $x \in X$. To compute entries of C, first note that when $x_1 \neq x_2$ and $i \neq j$ we have

$$C((x_1, i), (x_2, j)) = \frac{|\{x' \in X' : \{x_1, x_2\} \subset N(x')\}| \cdot (l-d)!}{d^2 (l-1)!} \cdot \frac{(l-2)!}{(l-d)!} = \frac{B(x_1, x_2)}{l-1}. \tag{52}$$

Finally, observe the following special cases:

- $C((x, i), (x, i)) = \frac{1}{d}$.
- $C((x_1, i), (x_2, i)) = 0$ if $x_1 \neq x_2$.
- $C((x, i), (x, j)) = 0$ when $i \neq j$.

Let $\widetilde{B} = B - \frac{1}{d}\,\mathrm{Id}$ and $\widetilde{C} = C - \frac{1}{d}\,\mathrm{Id}$. Let $\widetilde{J} = \frac{1}{l-1}(J - \mathrm{Id})$ be the $l \times l$ matrix that is $(l-1)^{-1}$ in the off-diagonal entries, and 0 along the diagonal. Then it is easy to see that

$$\widetilde{C} = \widetilde{B} \otimes \widetilde{J}. \tag{53}$$

The matrix $\widetilde{J}$ has a single eigenvalue equal to 1 and $l-1$ eigenvalues equal to $-\frac{1}{l-1}$, and $\widetilde{B}$ has a single eigenvalue equal to $1-1/d$ and the remaining are in the range $[-\frac{1}{d}, \lambda_M^2 - \frac{1}{d}]$. It follows that the top eigenvalue of $C = \widetilde{C} + \frac{1}{d}\,\mathrm{Id}$ is 1 (as expected) and the next one satisfies

$$\lambda_{\tilde{M}} \leq \max\left\{\lambda_M, \sqrt{\frac{1}{d(l-1)} + \frac{1}{d}}\right\},$$

which is bounded by $\max\left\{\lambda_M, \frac{1}{\sqrt{d-1}}\right\}$ since $l \geq d$. ◀

7 Weak Fortification of Entangled Games

Our goal in this section is to prove the following.

▶ **Theorem 34** (Theorem 3 restated). *Let $G' = (M \circ G \circ P)$ be a concatenated game obtained by concatenating two sides of a game G with some λ-spectral expanders M and P. If $\lambda \leq \frac{\varepsilon^2\delta}{56}$, then G' is (ε, δ) weakly-fortified against entangled substrategies.*

We need some basic matrix analytic facts.

7.1 Basic lemmas

Choi-Jamiolkowski isomorphism

We make use of the correspondence between bipartite states $|\psi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$ and linear operators $L : \mathcal{H}_2^* \to \mathcal{H}_1$ given by the Choi-Jamiolkowski isomorphism. Explicitly, let $|\psi\rangle \in \mathbb{C}^d \otimes \mathbb{C}^d$ be a quantum state and consider a Schmidt basis for $|\psi\rangle$ so we have $|\psi\rangle = \sum_{i=1}^d \sqrt{\lambda_i}|i\rangle|i\rangle$ where $\lambda_i \in \mathbb{R}^{\geq 0}$, up to a local change of basis. Set

$$\rho := \sum_{i=1}^d \lambda_i |i\rangle\langle i|. \tag{54}$$

▶ **Proposition 35.** *Let Z, W be two linear operators acting on $\mathbb{C}^d$ and let $|\psi\rangle$ and ρ be as above. Then,*

$$\langle\psi|Z \otimes W|\psi\rangle = \mathrm{Tr}(Z\rho^{1/2}W^T\rho^{1/2}).$$

Proof. Both expressions evaluate to $\sum_{i,j=1}^d \sqrt{\lambda_i\lambda_j}\, Z_{ij} \cdot W_{ij}$. ◀

For a density matrix ρ and a matrix A for convenience we sometimes denote $\mathrm{Tr}(A\rho)$ by $\mathrm{Tr}_\rho(A)$.

Matrix norms and inequalities

The Frobenius norm of a matrix $A \in \mathbb{C}^{n\times m}$ is defined as $\|A\|_F = \sqrt{\mathrm{Tr}(AA^\dagger)}$. The trace norm is defined as $\|A\|_{tr} = \mathrm{Tr}\sqrt{AA^\dagger}$. The following analogue of Proposition 18 will be used repeatedly in our argument.

▶ **Claim 36.** *Let M be a bipartite λ-spectral expander on vertex set $X' \cup X$. Let $\{A_{x'}\}_{x'\in X'}$ and ρ be positive semidefinite matrices. For all $x \in X$, define $A_x = \mathbb{E}_{x'\sim N(x)} A_{x'}$ and define $A = \mathbb{E}_{x\sim\mu} A_x$. Then*

$$\mathop{\mathbb{E}}_{x\sim\mu} \mathrm{Tr}_\rho((A_x - A)^2) \leq 2\lambda^2 \cdot \mathop{\mathbb{E}}_{x'\sim\mu'} \mathrm{Tr}_\rho(A_{x'}^2). \tag{55}$$

Proof. Define $S_{x'} = \rho^{1/2} A_{x'}$, $S_x = \rho^{1/2} A_x$, and $S = \mathbb{E}_x S_x = \rho^{1/2} A$. Using that M is a bipartite λ-spectral expander, for any fixed entry (i,j)

$$\mathop{\mathbb{E}}_{x} |(S_x)_{ij} - S_{ij}|^2 \leq \lambda^2 \cdot \mathop{\mathbb{E}}_{x'} |(S_{x'})_{ij} - S_{ij}|^2 \leq 2\lambda^2 \cdot \mathop{\mathbb{E}}_{x'} |(S_{x'})_{ij}|^2 \tag{56}$$

Summing over all entries,

$$\mathop{\mathbb{E}}_{x} \sum_{i,j} |(S_x)_{ij} - S_{ij}|^2 = \mathop{\mathbb{E}}_{x} \|S_x - S\|_F^2 \leq 2\lambda^2 \mathop{\mathbb{E}}_{x'} \sum_{i,j} |(S_{x'})_{ij}|^2 = 2\lambda^2 \mathop{\mathbb{E}}_{x'} \|S_{x'}\|_F^2. \tag{57}$$

Observing that $\mathrm{Tr}_\rho((A_x - A)^2) = \|S_x - S\|_F^2$ and $\|S_{x'}\|_F^2 = \mathrm{Tr}_\rho(A_{x'}^2)$, we obtain the desired result. ◀

If A has singular value decomposition $A = UJV^\dagger$ its pseudo-inverse is $A^{-1} = VJ^{-1}U^\dagger$, where J^{-1} is obtained from J by taking the reciprocal of non-zero diagonal entries. A simple consequence of the singular value decomposition is the following:

▶ **Fact 37.** *Let A be an $n \times n$ matrix. Then there exists a unitary matrix $\mathcal{U}$ such that $\mathcal{U}A$ is positive semi-definite.*

Proof. Write the SVD as $A = UJV^\dagger$, and choose $\mathcal{U} = VU^\dagger$. ◀

We make frequent use of the matrix Cauchy-Schwarz inequality.

▶ **Proposition 38.** *For any two matrices S, T we have*

$$\mathrm{Tr}(ST^\dagger) \leq \mathrm{Tr}(SS^\dagger)^{1/2} \cdot \mathrm{Tr}(TT^\dagger)^{1/2} = \|S\|_F \|T\|_F.$$

If S and T are Hermitian,

$$\mathrm{Tr}(STST) \leq \mathrm{Tr}(S^2T^2).$$

Finally, we need a variant of Powers-Størmer inequality due to Kittaneh [26]. This also played a role in the analysis of [16].

▶ **Lemma 39** ([26])**.** *Let X, Y be positive semidefinite matrices. Then*

$$\mathrm{Tr}\left((X-Y)^4\right) \leq \mathrm{Tr}\left((X^2 - Y^2)^2\right).$$

7.2 Proof of Theorem 3

At a high level, the proof of Theorem 3 follows the same outline as the classical proof of Section 5.

Consider a substrategy $\{A_{x'}^{a'}\}_{(x',a') \in X' \times A'}$, $\{B_{y'}^{b'}\}_{(y',b') \in Y' \times B'}$ for G'. Define $A_x = \mathbb{E}_{x' \sim N(x')} A_{x'}$ and $B_y = \mathbb{E}_{y' \sim N(y')} B_{y'}$.[13] Define $A = \mathbb{E}_{x \sim \mu_X} A_x$ and $B = \mathbb{E}_{y \sim \mu_Y} B_y$. To prove Theorem 3 we must analyze the following expression:

$$\begin{aligned}
\textsc{Val}^*(G', \{A_{x'}^{a'}\}, \{B_{y'}^{b'}\}) &= \mathop{\mathbb{E}}_{(x,y)\sim\mu} \mathop{\mathbb{E}}_{\substack{x' \sim N(x) \\ y' \sim N(y)}} \sum_{a',b'} V(a'_{x'}(x), b'_{y'}(y), x, y) \cdot \mathrm{Tr}(A_{x'}^{a'} \rho^{1/2} B_{y'}^{b'} \rho^{1/2}) \\
&= \mathop{\mathbb{E}}_{(x,y)\sim\mu} \sum_{a,b} V(a,b,x,y) \cdot \mathrm{Tr}(A_x^a \rho^{1/2} B_y^b \rho^{1/2}) \\
&= \mathop{\mathbb{E}}_{(x,y)\sim\mu} \mathrm{Tr}(A_x \rho^{1/2} B_y \rho^{1/2}) \cdot \sum_{V(a,b,x,y)=1} \frac{\mathrm{Tr}(A_x^a \rho^{1/2} B_y^b \rho^{1/2})}{\mathrm{Tr}(A_x \rho^{1/2} B_y \rho^{1/2})},
\end{aligned}$$

[13] In what follows, we assume without loss of generality that all $A_{x'}$ and $B_{y'}$ are invertible. Note that proving Theorem 3 for this subset of substrategies suffices. This follows by a limiting argument because of the continuity of (26) in $A_{x'}$ and $B_{y'}$.

where A_x^a and B_y^b are defined as in (11), and we use the convention that $0/0 = 0$. Our analysis splits into two cases. First let us consider the *small case*. This is handled by the following proposition.

▶ **Proposition 40.** *Suppose* $\mathrm{Tr}(\rho^{1/2}A\rho^{1/2}B) < \delta/2$. *Then* $\textsc{Val}^*(G', \{A_{x'}^{a'}\}, \{B_{y'}^{b'}\}) < \delta$.

Proof. First of all we have

$$\textsc{Val}^*(G', \{A_{x'}^{a'}\}, \{B_{y'}^{b'}\}) = \mathop{\mathbb{E}}_{x,y} \sum_{V(a,b,x,y)=1} \mathrm{Tr}(A_x^a \rho^{1/2} B_y^b \rho^{1/2}) \leq \mathop{\mathbb{E}}_{x,y} \mathrm{Tr}(A_x \rho^{1/2} B_y \rho^{1/2}).$$

Subtracting $\mathrm{Tr}(A\rho^{1/2}B\rho^{1/2})$,

$$\mathop{\mathbb{E}}_{x,y} \mathrm{Tr}(A_x \rho^{1/2} B_y \rho^{1/2}) - \mathrm{Tr}(A\rho^{1/2}B\rho^{1/2}) = \mathop{\mathbb{E}}_{x,y} \mathrm{Tr}((A_x - A)\rho^{1/2}(B_y - B)\rho^{1/2}).$$

By applying Cauchy-Schwarz to the latter expression and using Claim 36 it follows that

$$\textsc{Val}^*(G', \{A_{x'}^{a'}\}, \{B_{y'}^{b'}\}) \leq \delta/2 + \lambda^2;$$

this is smaller than δ by the choice of λ. ◀

The large case.

In this case, the hypothesis of Proposition 40 is not satisfied and without loss of generality we assume that

$$\min\left\{ \mathrm{Tr}_\rho(A), \mathrm{Tr}_\rho(B) \right\} \geq \mathrm{Tr}(A\rho^{1/2}B\rho^{1/2}) \geq \delta/2. \tag{58}$$

Let

$$\gamma := \mathop{\mathbb{E}}_{(x,y)\sim\mu} \mathrm{Tr}(A_x \rho^{1/2} B_y \rho^{1/2}). \tag{59}$$

By the triangle inequality,

$$\textsc{Val}^*(G', A_{x'}, B_{y'}) \leq \mathop{\mathbb{E}}_{(x,y)\sim\mu} |\,\mathrm{Tr}(A_x \rho^{1/2} B_y \rho^{1/2}) - \gamma| \tag{60}$$

$$+ \gamma \cdot \mathop{\mathbb{E}}_{(x,y)\sim\mu} \sum_{V(a,b,x,y)=1} \frac{\mathrm{Tr}(A_x^a \rho^{1/2} B_y^b \rho^{1/2})}{\mathrm{Tr}(A_x \rho^{1/2} B_y \rho^{1/2})}. \tag{61}$$

To bound the first term, we use the triangle inequality to get

$$|\,\mathrm{Tr}(A_x \rho^{1/2} B_y \rho^{1/2}) - \gamma| \leq |\,\mathrm{Tr}(A_x \rho^{1/2} B_y \rho^{1/2} - A\rho^{1/2}B\rho^{1/2})| + |\,\mathrm{Tr}(A\rho^{1/2}B\rho^{1/2}) - \gamma|. \tag{62}$$

The first term on the right-hand side of (62) can be bounded as

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(x,y)\sim\mu} |\,\mathrm{Tr}(A_x \rho^{1/2} B_y \rho^{1/2} - A\rho^{1/2}B\rho^{1/2})| \\
&\leq \mathop{\mathbb{E}}_{(x,y)\sim\mu} |\,\mathrm{Tr}(A_x \rho^{1/2} (B_y - B)\rho^{1/2})| + \mathop{\mathbb{E}}_{x} |\,\mathrm{Tr}((A_x - A)\rho^{1/2}B\rho^{1/2})| \\
&\leq \mathop{\mathbb{E}}_{(x,y)\sim\mu} \left[\mathrm{Tr}_\rho(A_x^2)^{1/2} \cdot \mathrm{Tr}_\rho((B_y - B)^2)^{1/2} \right] + \mathop{\mathbb{E}}_{x} \left[\mathrm{Tr}_\rho(B^2)^{1/2} \cdot \mathrm{Tr}_\rho((A_x - A)^2)^{1/2} \right] \\
&\leq \left(\mathop{\mathbb{E}}_{x} \mathrm{Tr}_\rho(A_x^2) \right)^{1/2} \cdot \left(\mathop{\mathbb{E}}_{y} \mathrm{Tr}_\rho((B_y - B)^2) \right)^{1/2} + \mathrm{Tr}_\rho(B^2)^{1/2} \cdot \left(\mathop{\mathbb{E}}_{x} \mathrm{Tr}_\rho((A_x - A)^2) \right)^{1/2} \\
&\leq 4 \cdot \lambda,
\end{aligned} \tag{63}$$

where the first inequality is the triangle inequality, the next two follow from Cauchy-Schwarz, and the last from Claim 36 and the trivial bounds $\mathrm{Tr}_\rho(A_x^2), \mathrm{Tr}_\rho(B^2) \leq \mathrm{Tr}(\rho) = 1$. To bound the second term on the right-hand side of (62) we note that

$$\begin{aligned}|\mathrm{Tr}(A\rho^{1/2}B\rho^{1/2}) - \gamma| &= |\mathop{\mathbb{E}}_{(x,y)\sim\mu} \mathrm{Tr}((A_x - A)\rho^{1/2}(B_y - B)\rho^{1/2})| \\ &\leq (\mathop{\mathbb{E}}_x \mathrm{Tr}_\rho[(A_x - A)^2])^{1/2} \cdot (\mathop{\mathbb{E}}_y \mathrm{Tr}_\rho[(B_y - B)^2])^{1/2},\end{aligned}$$

and the latter is again bounded by 2λ by Claim 36. In total we have

$$\mathop{\mathbb{E}}_{(x,y)\sim\mu} |\mathrm{Tr}(A_x\rho^{1/2}B_y\rho^{1/2}) - \gamma| \leq 4\lambda + 2\lambda^2 \leq \delta, \tag{64}$$

which provides an upper bound on the first term in the right-hand side of (60).

To bound the second term term in the right-hand side of (60) we use a strategy inspired in part by the parallel repetition theorem of [16]. Let U_x, V_y, U, V be a family of unitaries such that the operators

$$\Lambda_x = U_x\sqrt{A_x}\rho^{1/4}, \quad \Lambda = U\sqrt{A}\rho^{1/4}, \quad \Gamma_y = V_y\sqrt{B_y}\rho^{1/4}, \quad \Gamma = V\sqrt{B}\rho^{1/4} \tag{65}$$

are all positive semidefinite, which is possible by Fact 37. Note that this in particular implies that $\Lambda_x = \Lambda_x^\dagger$ and hence

$$\Lambda_x^2 = \Lambda_x^\dagger\Lambda_x = \rho^{1/4}\sqrt{A_x}U_x^\dagger U_x\sqrt{A_x}\rho^{1/4} = \rho^{1/4}A_x\rho^{1/4}, \tag{66}$$

and similarly $\Lambda^2 = \rho^{1/4}A\rho^{1/4}$, $\Gamma^2 = \rho^{1/4}B\rho^{1/4}$ and so on.

Define "rescaled" strategies by

$$\widehat{A_x^a} = U_x A_x^{-1/2}A_x^a A_x^{-1/2}U_x^\dagger, \quad \widehat{B_y^b} = V_y B_y^{-1/2}B_y^b B_y^{-1/2}V_y^\dagger, \tag{67}$$

where A_x^{-1}, B_y^{-1}'s are the pseudo-inverses of A_x, B_y. Note that the operators (67) satisfy $\widehat{A_x} = \sum_a \widehat{A_x^a}$, $\widehat{B_y} = \sum_b \widehat{B_y^b} \leq \mathrm{Id}$ as required. Let

$$K_{xy} = \frac{U_x A_x^{1/2}\rho^{1/2}B_y^{1/2}V_y^\dagger}{\sqrt{\mathrm{Tr}(A_x\rho^{1/2}B_y\rho^{1/2})}}, \qquad K = \frac{UA^{1/2}\rho^{1/2}B^{1/2}V^\dagger}{\sqrt{\mathrm{Tr}(A\rho^{1/2}B\rho^{1/2})}}. \tag{68}$$

By definition of $\Lambda_x, \Gamma_y, X, Y$ we see that the above is equivalent to

$$K_{xy} = \frac{\Lambda_x\Gamma_y}{\sqrt{\mathrm{Tr}(\Lambda_x^2\Gamma_y^2)}}, \qquad K = \frac{\Lambda\Gamma}{\sqrt{\mathrm{Tr}(\Lambda^2\Gamma^2)}}. \tag{69}$$

Now note the following identity

$$\frac{\mathrm{Tr}(A_x^a\rho^{1/2}B_y^b\rho^{1/2})}{\mathrm{Tr}(A_x\rho^{1/2}B_y\rho^{1/2})} = \mathrm{Tr}(\widehat{A_x^a}K_{xy}\widehat{B_y^b}K_{xy}^\dagger). \tag{70}$$

So to finish the argument it suffices to estimate

$$\mathop{\mathbb{E}}_{(x,y)\sim\mu} \sum_{V(a,b,x,y)=1} \mathrm{Tr}(\widehat{A_x^a}K_{xy}\widehat{B_y^b}K_{xy}^\dagger). \tag{71}$$

To this end note that since $\mathrm{Tr}(KK^\dagger) = 1$ it follows from the definition of $\textsc{Val}^*(G)$ that

$$\mathop{\mathbb{E}}_{(x,y)\sim\mu} \sum_{V(a,b,x,y)=1} \mathrm{Tr}(K\widehat{A_x^a}K^\dagger\widehat{B_y^b}) \leq \textsc{Val}^*(G). \tag{72}$$

To conclude we use the following proposition.

▶ **Proposition 41.** *Let K_{xy} and K be as above. Then*

$$\mathop{\mathbb{E}}_{(x,y)\sim\mu} \|K_{xy} - K\|_F^2 \leq \frac{12\lambda}{\delta}. \tag{73}$$

Before proving the proposition let us see how it implies the desired bound on the second term of (60).

$$\begin{aligned}
&|\operatorname{Tr}(K_{xy}\widehat{A_x^a}K_{xy}^\dagger\widehat{B_y^b}) - \operatorname{Tr}(K\widehat{A_x^a}K^\dagger\widehat{B_y^b})| \\
&\qquad \leq |\operatorname{Tr}((K_{xy}-K)\widehat{A_x^a}K_{xy}^\dagger\widehat{B_y^b})| + |\operatorname{Tr}(K\widehat{A_x^a}(K_{xy}^\dagger - K^\dagger)\widehat{B_y^b})| \\
&\qquad \leq \operatorname{Tr}((K_{xy}-K)\widehat{A_x^a}(K_{xy}-K)^\dagger\widehat{B_y^b})^{1/2}\cdot\operatorname{Tr}(K_{xy}\widehat{A_x^a}K_{xy}^\dagger\widehat{B_y^b})^{1/2} \\
&\qquad\quad + \operatorname{Tr}((K_{xy}-K)\widehat{A_x^a}(K_{xy}-K)^\dagger\widehat{B_y^b})^{1/2}\cdot\operatorname{Tr}(K\widehat{A_x^a}K^\dagger\widehat{B_y^b})^{1/2}.
\end{aligned} \tag{74}$$

Averaging with $\mathbb{E}_{(x,y)\sim\mu}\sum_{V(a,b,x,y)=1}$ and applying Cauchy-Schwarz we see that (74) is bounded by

$$\begin{aligned}
&\Big[\mathop{\mathbb{E}}_{(x,y)\sim\mu}\sum_{V(a,b,x,y)=1}\operatorname{Tr}((K_{xy}-K)\widehat{A_x^a}(K_{xy}-K)^\dagger\widehat{B_y^b})\Big]^{1/2}\cdot \\
&\Big[\Big(\mathop{\mathbb{E}}_{(x,y)\sim\mu}\sum_{V(a,b,x,y)=1}\operatorname{Tr}(K_{xy}\widehat{A_x^a}K_{xy}^\dagger\widehat{B_y^b})\Big)^{1/2} + \Big(\mathop{\mathbb{E}}_{(x,y)\sim\mu}\sum_{V(a,b,x,y)=1}\operatorname{Tr}(K\widehat{A_x^a}K^\dagger\widehat{B_y^b})\Big)^{1/2}\Big].
\end{aligned} \tag{75}$$

We claim that the second term in brackets is at most 2. To see this note that replacing the sum from $\sum_{V(a,b,x,y)=1}$ to a $\sum_{a,b}$ only increase the term, and the claim follows from $\operatorname{Tr}(K_{xy}K_{xy}^\dagger) = \operatorname{Tr}(KK^\dagger) = 1$. To bound the first term in (75), we again relax the summation from $\sum_{V(a,b,x,y)=1}$ to $\sum_{a,b}$. This is valid because all the operators of the form $(K_{xy} - K)\widehat{A_x^a}(K_{xy}-K)^\dagger, B_y \geq 0$ and hence all the additional terms introduced in the sum are nonnegative. The desired result follows because

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(x,y)\sim\mu}\sum_{a,b}\operatorname{Tr}((K_{xy}-K)\widehat{A_x^a}(K_{xy}-K)^\dagger\widehat{B_y^b}) \\
&\leq \mathop{\mathbb{E}}_{(x,y)\sim\mu}\operatorname{Tr}((K_{x,y}-K)(K_{x,y}-K)^\dagger) \\
&= \mathop{\mathbb{E}}_{x,y}\|K_{x,y}-K\|_F^2,
\end{aligned}$$

which is bounded by Proposition 41. Combining all bounds, from (74) we get

$$\begin{aligned}
\mathop{\mathbb{E}}_{(x,y)\sim\mu}\sum_{V(a,b,x,y)=1}\frac{\operatorname{Tr}(A_x^a\rho^{1/2}B_y^b\rho^{1/2})}{\operatorname{Tr}(A_x\rho^{1/2}B_y\rho^{1/2})} &= \mathop{\mathbb{E}}_{(x,y)\sim\mu}\sum_{V(a,b,x,y)=1}\operatorname{Tr}(\widehat{A_x^a}K_{xy}\widehat{B_y^b}K_{xy}^\dagger) \\
&\leq \mathop{\mathbb{E}}_{(x,y)\sim\mu}\sum_{V(a,b,x,y)=1}\operatorname{Tr}(\widehat{A_x^a}K\widehat{B_y^b}K^\dagger) \\
&\quad + |\operatorname{Tr}(K_{xy}\widehat{A_x^a}K_{xy}^\dagger\widehat{B_y^b}) - \operatorname{Tr}(K\widehat{A_x^a}K^\dagger\widehat{B_y^b})| \\
&\leq \textsc{Val}^*(G) + 2\cdot\left(\mathop{\mathbb{E}}_{(x,y)\sim\mu}\|K_{xy}-K\|_F^2\right)^{1/2} \\
&\leq \textsc{Val}^*(G) + 2\sqrt{\frac{12\lambda}{\delta}}.
\end{aligned}$$

The latter is bounded by ε by the choice of λ. It only remains to prove Proposition 41.

Proof of Proposition 41. We have

$$\|K_{xy} - K\|_F \leq \left\| \frac{\Lambda_x \Gamma_y}{\sqrt{\mathrm{Tr}(\Lambda_x^2 \Gamma_y^2)}} - \frac{\Lambda\Gamma}{\sqrt{\mathrm{Tr}(\Lambda^2\Gamma^2)}} \right\|_F + \left\| \frac{\Lambda_x \Gamma_y}{\sqrt{\mathrm{Tr}(\Lambda^2\Gamma^2)}} - \frac{\Lambda\Gamma}{\sqrt{\mathrm{Tr}(\Lambda^2\Gamma^2)}} \right\|_F. \quad (76)$$

For the first term,

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(x,y)\sim\mu} \left\| \frac{\Lambda_x \Gamma_y}{\sqrt{\mathrm{Tr}(\Lambda_x^2 \Gamma_y^2)}} - \frac{\Lambda_x \Gamma_y}{\sqrt{\mathrm{Tr}(\Lambda^2\Gamma^2)}} \right\|_F^2 \\
&= \mathop{\mathbb{E}}_{(x,y)\sim\mu} \mathrm{Tr}(\Lambda_x^2\Gamma_y^2) \cdot \left(\frac{1}{\sqrt{\mathrm{Tr}(\Lambda_x^2\Gamma_y^2)}} - \frac{1}{\sqrt{\mathrm{Tr}(\Lambda^2\Gamma^2)}} \right)^2 \\
&= \frac{1}{\mathrm{Tr}(\Lambda^2\Gamma^2)} \mathop{\mathbb{E}}_{(x,y)\sim\mu} \left(\sqrt{\mathrm{Tr}(\Lambda_x^2\Gamma_y^2)} - \sqrt{\mathrm{Tr}(\Lambda^2\Gamma^2)} \right)^2 \\
&\leq \frac{1}{\mathrm{Tr}(\Lambda^2\Gamma^2)} \mathop{\mathbb{E}}_{(x,y)\sim\mu} |\mathrm{Tr}(\Lambda_x^2\Gamma_y^2) - \mathrm{Tr}(\Lambda^2\Gamma^2)| \\
&\leq \frac{1}{\mathrm{Tr}(\Lambda^2\Gamma^2)} \mathop{\mathbb{E}}_{(x,y)\sim\mu} |\mathrm{Tr}((\Lambda_x^2 - \Lambda^2)\Gamma_y^2)| + |\mathrm{Tr}(\Lambda^2(\Gamma_y^2 - Y^2))| \\
&\leq \frac{1}{\mathrm{Tr}(\Lambda^2\Gamma^2)} \left| \mathop{\mathbb{E}}_x [\mathrm{Tr}((\Lambda_x^2 - \Lambda^2)^2)] \right|^{1/2} \cdot \left| \mathop{\mathbb{E}}_y [\mathrm{Tr}(\Gamma_y^4)] \right|^{1/2} \quad (77) \\
&+ \frac{1}{\mathrm{Tr}(\Lambda^2\Gamma^2)} \left| \mathop{\mathbb{E}}_x [\mathrm{Tr}((\Gamma_y^2 - \Gamma^2)^2)] \right|^{1/2} \cdot \mathrm{Tr}(\Lambda^4)^{1/2}, \quad (78)
\end{aligned}$$

where the last step follows from two applications of Cauchy-Schwarz. Rewriting the above in terms of A_x, B_y and ρ using (66) and its analogues we see that the term in (77) equals

$$\frac{1}{\mathrm{Tr}(A\rho^{1/2}B\rho^{1/2})} \left| \mathop{\mathbb{E}}_x [\mathrm{Tr}((A_x - A)\rho^{1/2}(A_x - A)\rho^{1/2})] \right|^{1/2} \cdot \left| \mathop{\mathbb{E}}_y [\mathrm{Tr}(B_y\rho^{1/2}B_y\rho^{1/2}) \right|^{1/2} \quad (79)$$

Bounding the last term $\mathrm{Tr}(B_y\rho^{1/2}B_y\rho^{1/2})$ by 1 and the first term by 2λ (which follows by applying Fact 38 and Claim 36) and doing the same analysis for (78) we see that

$$\mathop{\mathbb{E}}_{(x,y)\sim\mu} \left\| \frac{\Lambda_x \Gamma_y}{\sqrt{\mathrm{Tr}(\Lambda_x^2 \Gamma_y^2)}} - \frac{\Lambda_x \Gamma_y}{\sqrt{\mathrm{Tr}(\Lambda^2\Gamma^2)}} \right\|_F^2 \leq \frac{8\lambda}{\delta}. \quad (80)$$

To bound the second term in (76) we argue as follows:

$$\begin{aligned}
\|\Lambda_x\Gamma_y - \Lambda\Gamma\|_F^2 &\leq 2 \cdot \|(\Lambda_x - \Lambda)\Gamma_y\|_F^2 + 2 \cdot \|(\Gamma_y - \Gamma)X\|_F^2 \\
&= 2 \cdot \mathrm{Tr}(\Gamma_y^2(\Lambda_x - \Lambda)^2) + 2 \cdot \mathrm{Tr}((\Gamma_y - \Gamma)^2 X^2) \quad (81) \\
&\leq 2 \cdot \mathrm{Tr}(\Gamma_y^4)^{1/2} \cdot \mathrm{Tr}((\Lambda_x - \Lambda)^4)^{1/2} + 2 \cdot \mathrm{Tr}(\Lambda^4)^{1/2} \cdot \mathrm{Tr}((\Gamma_y - \Gamma)^4)^{1/2} \quad (82) \\
&\leq 2 \cdot \mathrm{Tr}(\Gamma_y^4)^{1/2} \cdot \mathrm{Tr}[(\Lambda_x^2 - \Lambda^2)^2]^{1/2} + 2 \cdot \mathrm{Tr}(\Lambda^4)^{1/2} \cdot \mathrm{Tr}[(\Gamma_y^2 - \Gamma^2)^2]^{1/2}, \quad (83)
\end{aligned}$$

where in the last step we used Lemma 39. Using the same bound on the above terms as in the above we see that

$$\mathop{\mathbb{E}}_{(x,y)\sim\mu} \|\Lambda_x\Gamma_y - \Lambda\Gamma\|_F^2 \leq 8\lambda. \quad (84)$$

Since in the large case $\mathrm{Tr}(A\rho^{1/2}B\rho^{1/2}) \geq \frac{\delta}{2}$ the result follows. ◀

8 Discussion and open problems

An obvious open problem is to extend our results to the case of multiplayer entangled games. This is likely to be achievable by combining the ideas of Sections 5.2 and 7. However, some subtleties arise with respect to the use of the Choi-Jamiolkowski isomorphism, and we leave this for future work.

An important (and somewhat surprising) message of our work is that there is a modified form of game concatenation with no adverse effect on the entangled value. This is notable because ordinary concatenation may appear to be not very well-behaved with respect to quantum strategies: we typically do not expect that entangled players would be able to answer a number of questions from a game G simultaneously, while preserving the same question/answer statistics as in G, as players' measurement operators associated with different questions generally do not commute.

The concatenation and composition of games play an important role in the classical setting in the context of PCPs [2, 14] and multiprover interactive proof systems [3]. It remains to be seen whether ideas related to our ordered fortification can be useful in lifting some of these techniques to the quantum world.

References

1 Rotem Arnon-Friedman, Renato Renner, and Thomas Vidick. Non-signalling parallel repetition using de finetti reductions. *arXiv preprint arXiv:1411.1582*, 2014.

2 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of np. *Journal of the ACM (JACM)*, 45(1):70–122, 1998.

3 László Babai, Lance Fortnow, and Carsten Lund. Nondeterministic exponential time has two-prover interactive protocols. In *Proceedings of Annual Symposium on Foundations of Computer Science (FOCS)*, pages 16–25, 1990.

4 Mohammad Bavarian, Thomas Vidick, and Henry Yuen. Anchoring games for parallel repetition. *arXiv preprint arXiv:1509.07466*, 2015.

5 Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *Proceedings of Symposium on Theory of computing (STOC)*, 1988.

6 Amey Bhangale, Ramprasad Saptharishi, Girish Varma, and Rakesh Venkat. On fortification of projection games. In *RANDOM (arXiv:1504.05556)*, 2015.

7 Yonatan Bilu and Nathan Linial. Lifts, discrepancy and nearly optimal spectral gap. *Combinatorica*, 26(5):495–519, 2006.

8 Mark Braverman and Ankit Garg. Small value parallel repetition for general games. In *Proceedings of the Annual ACM Symposium on Theory of Computing (STOC)*, pages 335–340, 2015.

9 Harry Buhrman, Serge Fehr, and Christian Schaffner. On the parallel repetition of multiplayer games: The no-signaling case. *arXiv preprint arXiv:1312.7455*, 2013.

10 André Chailloux and Giannicola Scarpa. Parallel repetition of entangled games with exponential decay via the superposed information cost. In *Proceeding of International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 296–307, 2014.

11 Kai-Min Chung, Xiaodi Wu, and Henry Yuen. Parallel repetition for entangled k-player games via fast quantum search. In *Proceedings of Conference on Computational Complexity (CCC)*, page 512, 2015.

12 John F Clauser, Michael A Horne, Abner Shimony, and Richard A Holt. Proposed experiment to test local hidden-variable theories. *Physical Review Letters*, 23(15):880–884, 1969.

13 Richard Cleve, Peter Høyer, Benjamin Toner, and John Watrous. Consequences and limits of nonlocal strategies. In *Proceedings Conference on Computational Complexity (CCC)*, pages 236–249, 2004.

14 Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM (JACM)*, 54(3):12, 2007.

15 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Proceedings ofAnnual ACM Symposium on Theory of Computing (STOC)*, pages 624–633, 2014.

16 Irit Dinur, David Steurer, and Thomas Vidick. A parallel repetition theorem for entangled projection games. In *Proceedings of Conference on Computational Complexity (CCC)*, pages 197–208, 2014.

17 Uriel Feige and Joe Kilian. Two-prover protocols—low error at affordable rates. *SIAM Journal on Computing*, 30(1):324–346, 2000.

18 Uriel Feige, Guy Kindler, and Ryan O'Donnell. Understanding parallel repetition requires understanding foams. In *Proceedings of Conference on Computational Complexity (CCC)*, pages 179–192, 2007.

19 Uriel Feige and Oleg Verbitsky. Error reduction by parallel repetition: a negative result. *Combinatorica*, 22(4):461–478, 2002.

20 Lance Fortnow, John Rompel, and Michael Sipser. On the power of multi-power interactive protocols. In *Proceedings of Structure in Complexity Theory Conference (CCC)*, pages 156–161, 1988.

21 Johan Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4), 2001.

22 Thomas Holenstein. Parallel repetition: simplifications and the no-signaling case. In *Proceedings of Symposium on Theory of computing (STOC)*, pages 411–419, 2007.

23 Tsuyoshi Ito and Thomas Vidick. A multi-prover interactive proof for NEXP sound against entangled provers. In *Proceedings of Foundations of Computer Science (FOCS)*, pages 243–252, 2012.

24 Rahul Jain, Attila Pereszlényi, and Penghui Yao. A parallel repetition theorem for entangled two-player one-round games under product distributions. In *Proceedings of Conference on Computational Complexity (CCC)*, pages 209–216, 2014.

25 Julia Kempe and Thomas Vidick. Parallel repetition of entangled games. In *Proceedings of Symposium on Theory of computing (STOC)*, pages 353–362, 2011.

26 Fuad Kittaneh. Inequalities for the schatten p-norm IV. *Communications in Mathematical Physics*, 106(4):581–585, 1986.

27 Cécilia Lancien and Andreas Winter. Parallel repetition and concentration for (sub-) no-signalling games via a flexible constrained de finetti reduction. *arXiv preprint arXiv:1506.07002*, 2015.

28 Dana Moshkovitz. Parallel repetition from fortification. In *Annual Symposium on Foundations of Computer Science (FOCS)*, pages 414–423, 2014.

29 Anup Rao. Parallel repetition in projection games and a concentration bound. *SIAM Journal on Computing*, 40(6):1871–1891, 2011.

30 Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

31 Ran Raz. A counterexample to strong parallel repetition. *SIAM Journal on Computing*, 40(3):771–777, 2011.

32 Thomas Vidick. Three-player entangled XOR games are NP-hard to approximate. In *Proceedings of the Foundations of Computer Science (FOCS)*, pages 766–775, 2013.

A Biregularization

In this section, we prove Lemmas 14 and 16. We start by the second lemma on the graphical games and then derive the general case by reduction.

Graphical games

Suppose G is a graphical game: there is a set of edges $E \subseteq X \times Y$ such that $\mu(x,y) = \frac{1}{|E|}$ for all $(x,y) \in E$. In this case we have

$$\mu(x) = \frac{|N(x)|}{|E|}, \qquad \mu(y) = \frac{|N(y)|}{|E|}, \qquad \forall x \in X,\, y \in Y. \tag{85}$$

Let $d_x := |N(x)|$ denote the degree of x, and set S_x be the set $\{x\} \times [d_x]$. Define

$$X_{int} = \bigcup_{x \sim X} S_x.$$

Note that $|X_{int}| = |E| \leq |X||Y|$. Define $M_{int}((x,i),x) = \frac{1}{d_x}$ for $i \in \{1,\dots,d_x\}$, and 0 otherwise. Construct Y_{int} and P_{int} similarly.

▶ **Proposition 42.** *Let $G_{int} = M_{int} \circ G \circ P_{int}$. Then marginal of μ_{int} induced on X_{int} and Y_{int} is uniform. Moreover, we have*

$$\textsc{Val}(G_{int}) = \textsc{Val}(G), \qquad \textsc{Val}^*(G_{int}) = \textsc{Val}^*(G). \tag{86}$$

Proof. It is easy to see that for all $X_{int} = (x,i) \in X_{int}$ we have $\mu_{int}(x_{int}) = \frac{1}{|E|}$ and similarly for all $y_{int} \in Y_{int}$. The claims $\textsc{Val}(G_{int}) = \textsc{Val}(G)$ and $\textsc{Val}^*(G_{int}) \leq \textsc{Val}^*(G)$ are true for all concatenated games in general. The final claim $\textsc{Val}^*(G_{int}) \geq \textsc{Val}^*(G)$ follows by considering the strategy $A_{(x,i)} = A_x$, $B_{(y,j)} = B_y$ which achieves the same value as (A_x, B_y) in G. ◀

General case

Although graphical games include many games considered in applications, it would nevertheless still be nice to extend the above construction to all games. We do not know how to do this exactly, but we can achieve an approximate variant.

The idea is essentially to approximate a general game by a graphical game. More formally, let $\tau \in (0,1)$ be an error parameter and q an integer such that $\frac{|E|}{\tau} \leq q \leq \frac{2|E|}{\tau}$. We have

$$\frac{\tau}{2|E|} \leq \frac{1}{q} \leq \frac{\tau}{|E|}. \tag{87}$$

We would like to define a game $\tilde{G}$ in which all probabilities in the underlying distribution $\tilde{\mu}(x,y)$ are fractions with denominator q. Let $\tilde{X} = X \cup \{x_{nul}\}$ and $\tilde{Y} = Y \cup \{y_{nul}\}$. For every $(x,y) \in X \times Y$ set

$$\tilde{\mu}(x,y) = \frac{\lfloor q \cdot \mu(x,y) \rfloor}{q}. \tag{88}$$

Finally let $\tilde{\mu}(x_{nul}, y_{nul})$ such that $\tilde{\mu}$ is a proper probability distribution (i.e. by transferring the excess probabilities to (x_{nul}, y_{nul})) and put an arbitrary winnable predicate on (x_{nul}, y_{nul}).

▶ **Proposition 43.** *The game $\tilde{G}$ is a graphical game with q (possibly parallel) edges. Moreover, we have*

$$\textsc{Val}(G) \leq \textsc{Val}(\tilde{G}) \leq \textsc{Val}(G) + \tau, \qquad \textsc{Val}^*(G) \leq \textsc{Val}^*(\tilde{G}) \leq \textsc{Val}^*(G) + \tau. \tag{89}$$

A few remarks are in order: firstly, since the previous construction for graphical games applies equally well in the presence of multiples edges, we can combine it with the above preprocessing to prove Lemma 14. Secondly, note that the operation $G \to \tilde{G}$ is value-increasing and hence preserves perfect completeness. Thirdly, note that the right scale for the error parameter τ is $\frac{c-s}{2}$ where $c - s$ is the completeness-soundness gap.

Proof. By construction all $\tilde{\mu}(x, y)$ are integer multiples of $\frac{1}{q}$. This ensures that the same is true for $\tilde{\mu}(x_{nul}, y_{nul})$. Since $\mu(x, y) \geq \tilde{\mu}(x, y)$ for all (x, y), for any strategy (f, g) for G we have

$$1 - \textsc{Val}(G, f, g) = \mathop{\mathbb{E}}_{(x,y)\sim\mu} \sum_{V(a,b,x,y)=0} f(x, a) \cdot g(y, b) \geq 1 - \textsc{Val}(\tilde{G}, f, g),$$

which shows that $\textsc{Val}(G) \leq \textsc{Val}(\tilde{G})$. For the other direction, consider an optimal strategy (f, g) for $\tilde{G}$ (which necessarily always wins on (x_{nul}, y_{nul})). We have,

$$\begin{aligned} 1 - \textsc{Val}(G) \leq 1 - \textsc{Val}(G, f, g) &= \mathop{\mathbb{E}}_{(x,y)\sim\mu} \sum_{V(a,b,x,y)=0} f(x, a) \cdot g(y, b) \\ &\leq \sum_{x,y} \tilde{\mu}(x, y) \sum_{V(a,b,x,y)=0} f(x, a) \cdot g(y, b) + \sum_{(x,y)\in E} (\mu(x, y) - \tilde{\mu}(x, y)) \\ &\leq 1 - \textsc{Val}(\tilde{G}) + \tau \end{aligned}$$

The quantum case is similar. ◀

The Classification of Reversible Bit Operations

Scott Aaronson[*1], Daniel Grier[†2], and Luke Schaeffer[3]

1 **University of Texas, Austin, USA**
aaronson@cs.utexas.edu

2 **Massachusets Institute of Technology, Cambridge, USA**
grierd@mit.edu

3 **Massachusets Institute of Technology, Cambridge, USA**
lrs@mit.edu

Abstract

We present a complete classification of all possible sets of classical reversible gates acting on bits, in terms of which reversible transformations they generate, assuming swaps and ancilla bits are available for free. Our classification can be seen as the reversible-computing analogue of *Post's lattice*, a central result in mathematical logic from the 1940s. It is a step toward the ambitious goal of classifying all possible quantum gate sets acting on qubits.

Our theorem implies a linear-time algorithm (which we have implemented), that takes as input the truth tables of reversible gates G and H, and that decides whether G generates H. Previously, this problem was not even known to be decidable (though with effort, one can derive from abstract considerations an algorithm that takes triply-exponential time). The theorem also implies that any n-bit reversible circuit can be "compressed" to an equivalent circuit, over the same gates, that uses at most $2^n \operatorname{poly}(n)$ gates and $O(1)$ ancilla bits; these are the first upper bounds on these quantities known, and are close to optimal. Finally, the theorem implies that every non-degenerate reversible gate can implement either every reversible transformation, or every affine transformation, when restricted to an "encoded subspace."

Briefly, the theorem says that every set of reversible gates generates either all reversible transformations on n-bit strings (as the Toffoli gate does); no transformations; all transformations that preserve Hamming weight (as the Fredkin gate does); all transformations that preserve Hamming weight mod k for some k; all affine transformations (as the Controlled-NOT gate does); all affine transformations that preserve Hamming weight mod 2 or mod 4, inner products mod 2, or a combination thereof; or a previous class augmented by a NOT or NOTNOT gate. Prior to this work, it was not even known that every class was finitely generated. Ruling out the possibility of additional classes, not in the list, requires involved arguments about polynomials, lattices, and Diophantine equations.

Due to the length of the proof, some parts of it have been omitted and may be found in the full version of the paper online.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases Reversible computation, Reversible gates, Circuit synthesis, Gate classification, Boolean logic, Post's lattice

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.23

* This work was done while the author was at MIT. Supported by an Alan T. Waterman Award from the National Science Foundation, under grant no. 1249349.
† Supported by an NSF Graduate Research Fellowship under grant no. 1122374.

© Scott Aaronson, Daniel Grier, and Luke Schaeffer;
licensed under Creative Commons License CC-BY

1 Introduction

The *pervasiveness of universality* – that is, the likelihood that a small number of simple operations already generate all operations in some relevant class – is one of the central phenomena in computer science. It appears, among other places, in the ability of simple logic gates to generate all Boolean functions (and of simple quantum gates to generate all unitary transformations); and in the simplicity of the rule sets that lead to Turing-universality, or to formal systems to which Gödel's theorems apply. Yet precisely because universality is so pervasive, it is often more interesting to understand the ways in which systems can *fail* to be universal.

In 1941, the great logician Emil Post [23] published a complete classification of all the ways in which sets of Boolean logic gates can fail to be universal: for example, by being monotone (like the AND and OR gates) or by being affine over $\mathbb{F}_2$ (like NOT and XOR). In universal algebra, closed classes of functions are known, somewhat opaquely, as *clones*, while the inclusion diagram of all Boolean clones is called *Post's lattice*. Post's lattice is surprisingly complicated, in part because Post did not assume that the constant functions 0 and 1 were available for free.[1]

This paper had its origin in our ambition to find the analogue of Post's lattice for all possible sets of *quantum* gates acting on qubits. We view this as a large, important, and underappreciated goal: something that could be to quantum computing theory almost what the Classification of Finite Simple Groups was to group theory. To provide some context, there are many sets of 1-, 2- and 3-qubit quantum gates that are known to be universal – either in the strong sense that they can be used to approximate any n-qubit unitary transformation to any desired precision, or in the weaker sense that they suffice to perform universal quantum computation (possibly in an encoded subspace). To take two examples, Barenco et al. [6] showed universality for the CNOT gate plus the set of all 1-qubit gates, while Shi [27] showed universality for the Toffoli and Hadamard gates.

There are also sets of quantum gates that are known *not* to be universal: for example, the basis-preserving gates, the 1-qubit gates, and most interestingly, the so-called *stabilizer gates* [12, 3] (that is, the CNOT, Hadamard, and $\pi/4$-Phase gates), as well as the stabilizer gates conjugated by 1-qubit unitary transformations[2]. What is *not* known is whether the preceding list basically exhausts the ways in which quantum gates on qubits can fail to be universal. Are there other elegant discrete structures, analogous to the stabilizer gates, waiting to be discovered? Are there any gate sets, other than conjugated stabilizer gates, that might give rise to intermediate complexity classes, neither contained in P nor equal to BQP?[3] How can we claim to understand quantum circuits – the bread-and-butter of quantum computing textbooks and introductory quantum computing courses – if we do not know the answers to such questions?

---

[1] If one *does* assume constants are free, then Post's lattice dramatically simplifies, with all non-universal gate sets either monotone or affine.

[2] In fact, Grier and Schaeffer [13] have extended our classification to these quantum stabilizer operations under the same model and with a similar proof structure. In the same paper, the authors classify the classical reversible transformations in the quantum regime (i.e., with quantum ancillas), heavily relying on the classification in this paper.

[3] To clarify, there are many restricted models of quantum computing known that are plausibly "intermediate" in that sense, including BosonSampling [1], the one-clean-qubit model [18], and log-depth quantum circuits [9]. However, with the exception of conjugated stabilizer gates, none of those models arises from simply considering which unitary transformations can be generated by some set of k-qubit gates. They all involve non-standard initial states, building blocks other than qubits, or restrictions on how the gates can be composed.

Unfortunately, working out the full "quantum Post's lattice" appears out of reach at present. This might surprise readers, given how much is known about particular quantum gate sets (e.g., those containing CNOT gates), but keep in mind that what is asked for is an accounting of *all* possibilities, no matter how exotic. Indeed, even classifying 1- and 2-qubit quantum gate sets remains wide open (!), and seems, without a new idea, to require studying the irreducible representations of thousands of groups. Recently, Aaronson and Bouland [2] completed a much simpler task, the classification of 2-mode beamsplitters; that was already a complicated undertaking.

Due to its length, this paper has been shortened for these proceedings. Its full version can be found on the arXiv [4].

1.1 Classical Reversible Gates

So one might wonder: can we at least understand all the possible sets of *classical reversible gates* acting on bits, in terms of which reversible transformations they generate? This is an obvious precursor to the quantum case, since every classical reversible gate is also a unitary quantum gate. But beyond that, the classical problem is extremely interesting in its own right, with (as it turns out) a rich algebraic and number-theoretic structure, and with many implications for reversible computing as a whole.

The notion of reversible computing [11, 29, 19, 8, 21, 24] arose from early work on the physics of computation, by such figures as Feynman, Bennett, Benioff, Landauer, Fredkin, Toffoli, and Lloyd. This community was interested in questions like: does universal computation inherently require the generation of entropy (say, in the form of waste heat)? Surprisingly, the theory of reversible computing showed that, in principle, the answer to this question is "no." *Deleting* information unavoidably generates entropy, according to *Landauer's principle* [19], but deleting information is not necessary for universal computation.

Formally, a reversible gate is just a permutation $G : \{0,1\}^k \rightarrow \{0,1\}^k$ of the set of k-bit strings, for some positive integer k. The most famous examples are:

- the 2-bit CNOT (Controlled-NOT) gate, which flips the second bit if and only if the first bit is 1;
- the 3-bit Toffoli gate, which flips the third bit if and only if the first two bits are both 1;
- the 3-bit Fredkin gate, which swaps the second and third bits if and only if the first bit is 1.

These three gates already illustrate some of the concepts that play important roles in this paper. The CNOT gate can be used to copy information in a reversible way, since it maps $x0$ to xx; and also to compute arbitrary affine functions over the finite field $\mathbb{F}_2$. However, because CNOT is *limited* to affine transformations, it is not computationally universal. Indeed, in contrast to the situation with irreversible logic gates, one can show that *no* 2-bit classical reversible gate is computationally universal. The Toffoli gate is computationally universal, because (for example) it maps $x, y, 1$ to $x, y, \overline{xy}$, thereby computing the NAND function. Moreover, Toffoli showed [29] – and we prove for completeness in Section 7.1 – that the Toffoli gate is universal in a stronger sense: it generates all possible reversible transformations $F : \{0,1\}^n \rightarrow \{0,1\}^n$ if one allows the use of ancilla bits, which must be returned to their initial states by the end.

But perhaps the most interesting case is that of the Fredkin gate. Like the Toffoli gate, the Fredkin gate is computationally universal: for example, it maps $x, y, 0$ to $x, \overline{x}y, xy$, thereby computing the AND function. But the Fredkin gate is *not* universal in the stronger sense. The reason is that it is *conservative*: that is, it never changes the total Hamming weight of

the input. Far from being just a technical issue, conservativity was regarded by Fredkin and the other reversible computing pioneers as a sort of discrete analogue of the conservation of energy – and indeed, it plays a central role in certain physical realizations of reversible computing (for example, billiard-ball models, in which the total number of billiard balls must be conserved).

However, all we have seen so far are three specific examples of reversible gates, each leading to a different behavior. To anyone with a mathematical mindset, the question remains: what are all the *possible* behaviors? For example: is Hamming weight the only possible "conserved quantity" in reversible computation? Are there other ways, besides being affine, to fail to be computationally universal? Can one *derive*, from first principles, why the classes of reversible transformations generated by CNOT, Fredkin, etc. are somehow special, rather than just pointing to the sociological fact that these are classes that people in the early 1980s happened to study?

1.2 Ground Rules

In this work, we achieve a complete classification of all possible sets of reversible gates acting on bits, in terms of which reversible transformations $F : \{0,1\}^n \rightarrow \{0,1\}^n$ they generate. Before describing our result, let us carefully explain the ground rules.

First, we assume that swapping bits is free. This simply means that we do not care how the input bits are labeled – or, if we imagine the bits carried by wires, then we can permute the wires in any way we like. The second rule is that an unlimited number of ancilla bits may be used, *provided* the ancilla bits are returned to their initial states by the end of the computation. This second rule might look unfamiliar, but in the context of reversible computing, it is the right choice.

We need to allow ancilla bits because if we do not, then countless transformations are disallowed for trivial reasons. (Restricting a reversible circuit to use *no* ancillas is like restricting a Turing machine to use no memory, besides the n bits that are used to write down the input.) We are forced to say that, although our gates might generate some reversible transformation $F(x,0) = (G(x),0)$, they do not generate the smaller transformation G. The exact value of n then also takes on undeserved importance, as we need to worry about "small-n effects": e.g., that a 3-bit gate cannot be applied to a 2-bit input.

As for the number of ancilla bits: it will *turn out*, because of our classification theorem, that every reversible gate needs only $O(1)$ ancilla bits[4] to generate every n-bit reversible transformation that it can generate at all. However, we do not wish to prejudge this question; if there had been reversible gates that could generate certain transformations, but only by using (say) 2^{2^n} ancilla bits, then that would have been fascinating to know. For the same reason, we do not wish prematurely to restrict the number of ancilla bits that can be 0, or the number that can be 1.

On the other hand, the ancilla bits must be returned to their original states because if they are not, then the computation was not really reversible. One can then learn something about the computation by examining the ancilla bits – if nothing else, then the fact that the computation was done at all. The symmetry between input and output is broken; one cannot then run the computation backwards without setting the ancilla bits differently. This is not just a philosophical problem: if the ancilla bits carry away information about the input

[4] Since it is easy to show that a constant number of ancilla bits are sometimes needed (see Proposition 9), this is the optimal answer, up to the value of the constant (which might depend on the gate set).

x, then *entropy*, or waste heat, has been leaked into the computer's environment. Worse yet, if the reversible computation is a subroutine of a quantum computation, then the leaked entropy will cause *decoherence*, preventing the branches of the quantum superposition with different x values from interfering with each other, as is needed to obtain a quantum speedup. In reversible computing, the technical term for ancilla bits that still depend on x after a computation is complete is *garbage*.[5]

1.3 Our Results

Even after we assume that bit swaps and ancilla bits are free, it remains a significant undertaking to work out the complete list of reversible gate classes, and (especially!) to prove that the list is complete. Doing so is this paper's main technical contribution.

We give a formal statement of the classification theorem in Section 3, and we show the lattice of reversible gate classes in Figure 3. For now, let us simply state the main conclusions informally.

1. **Conserved Quantities.** The following is the complete list of the "global quantities" that reversible gate sets can conserve (if we restrict attention to non-degenerate gate sets, and ignore certain complications caused by linearity and affineness): Hamming weight, Hamming weight mod k for any $k \geq 2$, and inner product mod 2 between pairs of inputs.
2. **Anti-Conservation.** There are gates, such as the NOT gate, that "anti-conserve" the Hamming weight mod 2 (i.e., always change it by a fixed nonzero amount). However, there are no analogues of these for any of the other conserved quantities.
3. **Encoded Universality.** In terms of their "computational power," there are only three kinds of reversible gate sets: degenerate (e.g., NOTs, bit-swaps), non-degenerate but affine (e.g., CNOT), and non-affine (e.g., Toffoli, Fredkin). More interestingly, every non-affine gate set can implement every reversible transformation, and every non-degenerate affine gate set can implement every affine transformation, *if* the input and output bits are encoded by longer strings in a suitable way. For details about "encoded universality," see Section 4.4.
4. **Sporadic Gate Sets.** The conserved quantities interact with linearity and affineness in complicated ways, producing "sporadic" affine gate sets that we have classified. For example, non-degenerate affine gates can preserve Hamming weight mod k, but only if $k = 2$ or $k = 4$. All gates that preserve inner product mod 2 are linear, and all linear gates that preserve Hamming weight mod 4 also preserve inner product mod 2. As a further complication, for an affine transformation $F(x) = Ax + b$, it is possible for A to be orthogonal, mod-2-preserving, or mod-4-preserving without F being orthogonal, mod-2-preserving, or mod-4-preserving, respectively.
5. **Finite Generation.** For each closed class of reversible transformations, there is a single gate that generates the entire class. (*A priori*, it is not even obvious that every class is finitely generated, or that there is "only" a countable infinity of classes!) For more, see Section 4.1.
6. **Symmetry.** Every reversible gate set is symmetric under interchanging the roles of 0 and 1. For more, see Section 4.1.

[5] In Section 2.3, we will discuss a modified rule, which allows a reversible circuit to change the ancilla bits, as long as they change in a way that is independent of the input x. We will show that this "loose ancilla rule" causes only a small change to our classification theorem.

1.4 Algorithmic and Complexity Aspects

Perhaps most relevant to theoretical computer scientists, our classification theorem leads to new algorithms and complexity results about reversible gates and circuits: results that follow easily from the classification, but that we have no idea how to prove otherwise.

Let RevGen (Reversible Generation) be the following problem: we are given as input the truth tables of reversible gates $G_1, \ldots, G_K$, as well as of a target gate H, and wish to decide whether the G_i's generate H. Then we obtain a linear-time algorithm for RevGen. Here, of course, "linear" means linear in the sizes of the truth tables, which is $n2^n$ for an n-bit gate. However, if just a tiny amount of "summary data" about each gate G is provided – namely, the possible values of $|G(x)| - |x|$, where $|\cdot|$ is the Hamming weight, as well as which affine transformation G performs if it is affine – then the algorithm actually runs in $O(n^\omega)$ time, where ω is the matrix multiplication exponent.

We have implemented this algorithm; code is available for download at [25]. For more details see Section 4.2.

Our classification theorem also implies the first general upper bounds (i.e., bounds that hold for all possible gate sets) on the number of gates and ancilla bits needed to implement reversible transformations. In particular, we show (see Section 4.3) that if a set of reversible gates generates an n-bit transformation F at all, then it does so via a circuit with at most $2^n \operatorname{poly}(n)$ gates and $O(1)$ ancilla bits. These bounds are close to optimal.

By contrast, let us consider the situation for these problems without the classification theorem. Suppose, for example, that we want to know whether a reversible transformation $H : \{0,1\}^n \to \{0,1\}^n$ can be synthesized using gates $G_1, \ldots, G_K$. If we knew some upper bound on the number of ancilla bits that might be needed by the generating circuit, then if nothing else, we could of course solve this problem by brute force. The trouble is that, without the classification, it is not obvious how to prove *any* upper bound on the number of ancillas – not even, say, Ackermann(n). This makes it unclear, *a priori*, whether RevGen is even *decidable*, never mind its complexity!

One *can* show on abstract grounds that RevGen is decidable, but with an astronomical running time. To explain this requires a short digression. In universal algebra, there is a body of theory (see e.g. [20]), which grew out of Post's original work [23], about the general problem of classifying closed classes of functions (clones) of various kinds. The upshot is that every clone is characterized by an *invariant* that all functions in the clone preserve: for example, affineness for the NOT and XOR functions, or monotonicity for the AND and OR functions. The clone can then be shown to contain *all* functions that preserve the invariant. (There is a formal definition of "invariant," involving polymorphisms, which makes this statement not a tautology, but we omit it.) Alongside the lattice of clones of functions, there is a dual lattice of *coclones* of invariants, and there is a Galois connection relating the two: as one adds more functions, one preserves fewer invariants, and vice versa.

In response to an inquiry by us, Emil Jeřábek recently showed [15] that the clone/coclone duality can be adapted to the setting of reversible gates. This means that we know, even without a classification theorem, that every closed class of reversible transformations is uniquely determined by the invariants that it preserves.

Unfortunately, this elegant characterization does not give rise to feasible algorithms. The reason is that, for an n-bit gate $G : \{0,1\}^n \to \{0,1\}^n$, the invariants could in principle involve all 2^n inputs, as well arbitrary polymorphisms mapping those inputs into a commutative monoid. Thus the number of polymorphisms one needs to consider grows at least like $2^{2^{2^n}}$. Now, the word problem for commutative monoids is decidable, by reduction to the ideal membership problem (see, e.g., [17, p. 55]). And by putting these facts together, one can

derive an algorithm for RevGen that uses doubly-exponential space and triply-exponential time, as a function of the truth table sizes: in other words, $\exp(\exp(\exp(\exp(n))))$ time, as a function of n. We believe it should also be possible to extract $\exp(\exp(\exp(\exp(n))))$ upper bounds on the number of gates and ancillas from this algorithm, although we have not verified the details.

1.5 Proof Ideas

We hope we have made the case that the classification theorem improves the complexity situation for reversible circuit synthesis! Even so, some people might regard classifying all possible reversible gate sets as a complicated, maybe worthwhile, but fundamentally tedious exercise. Can't such problems be automated via computer search? On the contrary, there are specific aspects of reversible computation that make this classification problem both unusually rich, and unusually hard to reduce to any finite number of cases.

We already discussed the astronomical number of possible invariants that even a tiny reversible gate (say, a 3-bit gate) might satisfy, and the hopelessness of enumerating them by brute force. However, even if we could cut down the number of invariants to something reasonable, there would still be the problem that the size, n, of a reversible gate can be arbitrarily large – and as one considers larger gates, one can discover more and more invariants. Indeed, that is precisely what happens in our case, since the Hamming weight mod k invariant can only be "noticed" by considering gates on k bits or more. There are also "sporadic" affine classes that can only be found by considering 6-bit gates.

Of course, it is not hard just to *guess* a large number of reversible gate classes (affine transformations, parity-preserving and parity-flipping transformations, etc.), prove that these classes are all distinct, and then prove that each one can be generated by a simple set of gates (e.g., CNOT or Fredkin + NOT). Also, once one has a sufficiently powerful gate (say, the CNOT gate), it is often straightforward to classify all the classes *containing* that gate. So for example, it is relatively easy to show that CNOT, together with any non-affine gate, generates all reversible transformations.

As usual with classification problems, the hard part is to rule out exotic additional classes: most of the work, one might say, is not about what is there, but about what isn't there. It is one thing to synthesize some random 1000-bit reversible transformation using only Toffoli gates, but quite another to synthesize a Toffoli gate using only the random 1000-bit transformation!

Thinking about this brings to the fore the central issue: that in reversible computation, it is not enough to output some desired string $F(x)$; one needs to output nothing else *besides* $F(x)$. And hence, for example, it does not suffice to look inside the random 1000-bit reversible gate G, to show that it contains a NAND gate, which is computationally universal. Rather, one needs to deal with *all* of G's outputs, and show that one can eliminate the undesired ones.

The way we do that involves another characteristic property of reversible circuits: that they can have "global conserved quantities," such as Hamming weight. Again and again, we need to prove that if a reversible gate G *fails* to conserve some quantity, such as the Hamming weight mod k, then that fact alone implies that we can use G to implement a desired behavior. This is where elementary algebra and number theory come in.

There are two aspects to the problem. First, we need to understand something about the possible quantities that a reversible gate can conserve. For example, we will need the following three results:

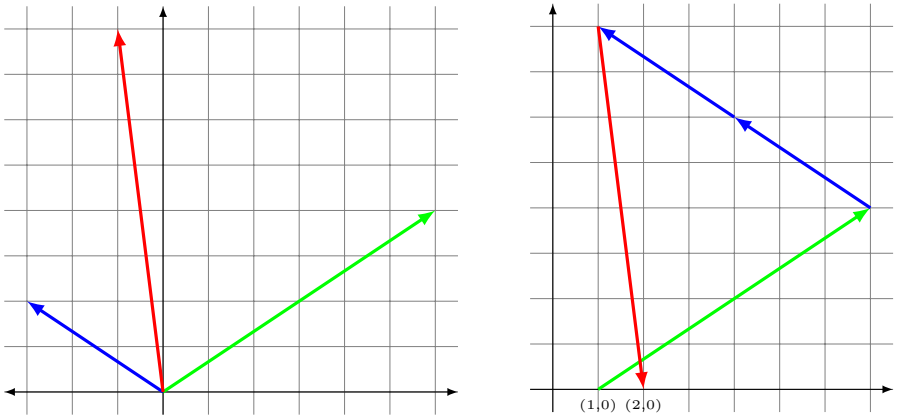


Figure 1 Moving within first quadrant of lattice to construct a COPY gate.

- No reversible gate can change the Hamming weight of some input and also conserve inner products mod k, unless $k = 2$.
- No reversible gate can change Hamming weight mod k by a fixed, nonzero amount, unless $k = 2$.
- No nontrivial linear gate can conserve Hamming weight mod k, unless $k = 2$ or $k = 4$.

We prove each of these statements in Section 6, using arguments based on complex polynomials.

Next, using our knowledge about the possible conserved quantities, we need procedures that take any gate G that fails to conserve some quantity, and that use G to implement a desired behavior (say, making a single copy of a bit, or changing an inner product by exactly 1). We then leverage that behavior to generate a desired gate (say, a Fredkin gate). The two core tasks turn out to be the following:

- Given any non-affine gate, we need to construct a Fredkin gate.
- Given any non-orthogonal linear gate, we need to construct a CNOTNOT gate, a parity-preserving version of CNOT that maps x, y, z to $x, y \oplus x, z \oplus x$.

These proofs are quite lengthy and are included in the full version of our paper [4]. The solution involves 3-dimensional lattices: that is, subsets of $\mathbb{Z}^3$ closed under integer linear combinations. We argue, in essence, that the only possible obstruction to the desired behavior is a "modularity obstruction," but the assumption about the gate G rules out such an obstruction.

We can illustrate this with an example that ends up *not* being needed in the final classification proof, but that we worked out earlier in this research.[6] Let G be any gate that does not conserve (or anti-conserve) the Hamming weight mod k for any $k \geq 2$, and suppose we want to use G to construct a CNOT gate.

[6] In general, after completing the classification proof, we were able to go back and simplify it substantially, by removing results – for example, about the generation of CNOT gates – that were important for working out the lattice in the first place, but which then turned out to be subsumed (or which *could* be subsumed, with modest additional effort) by later parts of the classification. Our current proof reflects these simplifications.

Then we examine how G behaves on restricted inputs: in this case, on inputs that consist entirely of some number of copies of x and $\overline{x}$, where $x \in \{0,1\}$ is a bit, as well as constant 0 and 1 bits. For example, perhaps G can increase the number of copies of x by 5 while decreasing the number of copies of $\overline{x}$ by 7, and can *also* decrease the number of copies of x by 6 without changing the number of copies of $\overline{x}$. Whatever the case, the set of possible behaviors generates some lattice: in this case, a lattice in $\mathbb{Z}^2$ (see Figure 1). We need to argue that the lattice contains a distinguished point encoding the desired "copying" behavior. In the case of the CNOT gate, the point is $(1,0)$, since we want one more copy of x and no more copies of $\overline{x}$. Showing that the lattice contains $(1,0)$, in turn, boils down to arguing that a certain system of Diophantine linear equations must have a solution. One can do this, finally, by using the assumption that G does not conserve or anti-conserve the Hamming weight mod k for any k.

To generate the Fredkin gate, we instead use the Chinese Remainder Theorem to combine gates that change the inner product mod p for various primes p into a gate that changes the inner product between two inputs by exactly 1; while to generate the CNOTNOT gate, we exploit the assumption that our generating gates are linear. In all these cases, it is crucial that we know, from Section 6, that certain quantities *cannot* be conserved by any reversible gate.

There are a few parts of the classification proof that basically *do* come down to enumerating cases, but we hope to have given a sense for the interesting parts.

1.6 Related Work

Surprisingly, the general question of classifying reversible gates such as Toffoli and Fredkin appears never to have been asked, let alone answered, prior to this work.

In the reversible computing literature, there are hundreds of papers on synthesizing reversible circuits (see [24] for a survey), but most of them focus on practical considerations: for example, trying to minimize the number of Toffoli gates or other measures of interest, often using software optimization tools. We found only a tiny amount of work relevant to the classification problem: notably, an unpublished preprint by Lloyd [21], which shows that every non-affine reversible gate is computationally universal, if one does not care what garbage is generated in addition to the desired output. Lloyd's result was subsequently rediscovered by Kerntopf et al. [16] and De Vos and Storme [30].

There is also work by Morita et al. [22] that uses brute-force enumeration to classify certain reversible computing elements with 2, 3, or 4 wires, but the notion of "reversible gate" there is very different from the standard one (the gates are for routing a single "billiard ball" element rather than for transforming bit strings, and they have internal state). Finally, there is work by Strazdins [28], not motivated by reversible computing, which considers classifying reversible Boolean functions, but which imposes a separate requirement on each output bit that it belong to one of the classes from Post's original lattice, and which thereby misses all the reversible gates that conserve "global" quantities, such as the Fredkin gate.[7]

[7] Because of different rules regarding constants, developed with Post's lattice rather than reversible computing in mind, Strazdins also includes classes that we do not (e.g., functions that always map 0^n or 1^n to themselves, but are otherwise arbitrary). To use our notation, his 13-class lattice ends up intersecting our infinite lattice in just five classes: $\langle\varnothing\rangle$, $\langle\text{NOT}\rangle$, $\langle\text{CNOTNOT}, \text{NOT}\rangle$, $\langle\text{CNOT}\rangle$, and $\langle\text{Toffoli}\rangle$.

2 Notation and Definitions

$\mathbb{F}_2$ means the field of 2 elements. $[n]$ means $\{1,\ldots,n\}$. We denote by $e_1,\ldots,e_n$ the standard basis for the vector space $\mathbb{F}_2^n$: that is, $e_1=(1,0,\ldots,0)$, etc.

Let $x=x_1\ldots x_n$ be an n-bit string. Then $\overline{x}$ means x with all n of its bits inverted. Also, $x\oplus y$ means bitwise XOR, x,y or xy means concatenation, x^k means the concatenation of k copies of x, and $|x|$ means the Hamming weight. The *parity* of x is $|x| \bmod 2$. The *inner product* of x and y is the integer $x\cdot y=x_1y_1+\cdots+x_ny_n$. Note that

$$x\cdot(y\oplus z)\equiv x\cdot y+x\cdot z\,(\mathrm{mod}\,2)\,,$$

but the above need not hold if we are not working mod 2.

By $\mathrm{gar}\,(x)$, we mean garbage depending on x: that is, "scratch work" that a reversible computation generates along the way to computing some desired function $f\,(x)$. Typically, the garbage later needs to be *uncomputed*. Uncomputing, a term introduced by Bennett [8], simply means running an entire computation in reverse, after the output $f\,(x)$ has been safely stored.

2.1 Gates

By a *(reversible) gate*, throughout this paper we will mean a reversible transformation G on the set of k-bit strings: that is, a permutation of $\{0,1\}^k$, for some fixed k. Formally, the terms 'gate' and 'reversible transformation' will mean the same thing; 'gate' just connotes a reversible transformation that is particularly small or simple.

A gate is *nontrivial* if it does something other than permute its input bits, and *non-degenerate* if it does something other than permute its input bits and/or apply NOT's to some subset of them.

A gate G is *conservative* if it satisfies $|G\,(x)|=|x|$ for all x. A gate is *mod-k-respecting* if there exists a j such that

$$|G\,(x)|\equiv|x|+j\,(\mathrm{mod}\,k)$$

for all x. It's *mod-k-preserving* if moreover $j=0$. It's *mod-preserving* if it's mod-k-preserving for some $k\geq 2$, and *mod-respecting* if it's mod-k-respecting for some $k\geq 2$.

As special cases, mod-2-respecting gates and mod-2-preserving gates are called is also called *parity-respecting* and *parity-preserving* respectively. A gate G such that

$$|G\,(x)|\not\equiv|x|\,(\mathrm{mod}\,2)$$

for all x is called *parity-flipping*. In Theorem 12, we will prove that parity-flipping gates are the *only* examples of mod-respecting gates that are not mod-preserving.

The *respecting number* of a gate G, denoted $k\,(G)$, is the largest k such that G is mod-k-respecting. (By convention, if G is conservative then $k\,(G)=\infty$, while if G is non-mod-respecting then $k\,(G)=1$.) We have the following fact:

▶ **Proposition 1.** *G is mod-ℓ-respecting if and only if ℓ divides $k\,(G)$.*

Proof. If ℓ divides $k\,(G)$, then certainly G is mod-ℓ-respecting. Now, suppose G is mod-ℓ-respecting but ℓ does not divide $k\,(G)$. Then G is both mod-ℓ-respecting and mod-$k(G)$-respecting. So by the Chinese Remainder Theorem, G is mod-$\mathrm{lcm}\,(\ell,k\,(G))$-respecting. But this contradicts the definition of $k\,(G)$. ◀

A gate G is *affine* if it implements an affine transformation over $\mathbb{F}_2$: that is, if there exists an invertible matrix $A \in \mathbb{F}_2^{k \times k}$, and a vector $b \in \mathbb{F}_2^k$, such that $G(x) = Ax \oplus b$ for all x. A gate is *linear* if moreover $b = 0$. A gate is *orthogonal* if it satisfies

$$G(x) \cdot G(y) \equiv x \cdot y \pmod 2$$

for all x, y. (We will observe, in Lemma 14, that every orthogonal gate is linear.) Also, if $G(x) = Ax \oplus b$ is affine, then the *linear part of* G is the linear transformation $G'(x) = Ax$. We call G orthogonal in its linear part, mod-k-preserving in its linear part, etc. if G' satisfies the corresponding invariant. A gate that is orthogonal in its linear part is also called an *isometry*.

Given two gates G and H, their *tensor product*, $G \otimes H$, is a gate that applies G and H to disjoint sets of bits. We will often use the tensor product to produce a single gate that combines the properties of two previous gates. Also, we denote by $G^{\otimes t}$ the tensor product of t copies of G.

2.2 Gate Classes

Let $S = \{G_1, G_2, \ldots\}$ be a set of gates, possibly on different numbers of bits and possibly infinite. Then $\langle S \rangle = \langle G_1, G_2, \ldots \rangle$, *the class of reversible transformations generated by* S, can be defined as the smallest set of reversible transformations $F : \{0,1\}^n \to \{0,1\}^n$ that satisfies the following closure properties:

1. **Base case.** $\langle S \rangle$ contains S, as well as the identity function $F(x_1 \ldots x_n) = x_1 \ldots x_n$ for all $n \geq 1$.
2. **Composition rule.** If $\langle S \rangle$ contains $F(x_1 \ldots x_n)$ and $G(x_1 \ldots x_n)$, then $\langle S \rangle$ also contains $F(G(x_1 \ldots x_n))$.
3. **Swapping rule.** If $\langle S \rangle$ contains $F(x_1 \ldots x_n)$, then $\langle S \rangle$ also contains all possible functions $\sigma\left(F\left(x_{\tau(1)} \ldots x_{\tau(n)}\right)\right)$ obtained by permuting F's input and output bits.
4. **Extension rule.** If $\langle S \rangle$ contains $F(x_1 \ldots x_n)$, then $\langle S \rangle$ also contains the function

   $$G(x_1 \ldots x_n, b) := (F(x_1 \ldots x_n), b),$$

   in which b occurs as a "dummy" bit.
5. **Ancilla rule.** If $\langle S \rangle$ contains a function F that satisfies

   $$F(x_1 \ldots x_n, a_1 \ldots a_k) = (G(x_1 \ldots x_n), a_1 \ldots a_k) \quad \forall x_1 \ldots x_n \in \{0,1\}^n,$$

   for some smaller function G and fixed "ancilla" string $a_1 \ldots a_k \in \{0,1\}^k$ that does not depend on x, then $\langle S \rangle$ also contains G. (Note that, if the a_i's are set to other values, then F need not have the above form.)

Note that because of reversibility, the set of n-bit transformations in $\langle S \rangle$ (for any n) always forms a group. Indeed, if $\langle S \rangle$ contains F, then clearly $\langle S \rangle$ contains all the iterates $F^2(x) = F(F(x))$, etc. But since there must be some positive integer m such that $F^m(x) = x$, this means that $F^{m-1}(x) = F^{-1}(x)$. Thus, we do not need a separate rule stating that $\langle S \rangle$ is closed under inverses.

We say S *generates* the reversible transformation F if $F \in \langle S \rangle$. We also say that S generates $\langle S \rangle$.

Given an arbitrary set $\mathcal{C}$ of reversible transformations, we call $\mathcal{C}$ a *reversible gate class* (or *class* for short) if $\mathcal{C}$ is closed under rules (1)-(5) above: in other words, if there exists an S such that $\mathcal{C} = \langle S \rangle$.

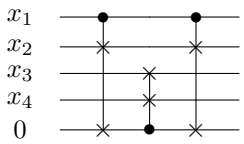


Figure 2 Generating a Controlled-Controlled-Swap gate from Fredkin.

A *reversible circuit* for the function F, over the gate set S, is an explicit procedure for generating F by applying gates in S, and thereby showing that $F \in \langle S \rangle$. An example is shown in Figure 2. Reversible circuit diagrams are read from left to right, with each bit that occurs in the circuit (both input and ancilla bits) represented by a horizontal line, and each gate represented by a vertical line.

If every gate $G \in S$ satisfies some invariant, then we can also describe S and $\langle S \rangle$ as satisfying that invariant. So for example, the set $\{\text{CNOTNOT}, \text{NOT}\}$ is affine and parity-respecting, and so is the class that it generates. Conversely, S violates an invariant if any $G \in S$ violates it.

Just as we defined the respecting number $k(G)$ of a gate, we would like to define the respecting number $k(S)$ of an entire gate set. To do so, we need a proposition about the behavior of $k(G)$ under tensor products.

▶ **Proposition 2.** *For all gates G and H,*

$$k(G \otimes H) = \gcd(k(G), k(H)).$$

Proof. Letting $\gamma = \gcd(k(G), k(H))$, clearly $G \otimes H$ is mod-γ-respecting. To see that $G \otimes H$ is not mod-ℓ-respecting for any $\ell > \gamma$: by definition, ℓ must fail to divide either $k(G)$ or $k(H)$. Suppose it fails to divide $k(G)$ without loss of generality. Then G cannot be mod-ℓ-respecting, by Proposition 1. But if we consider pairs of inputs to $G \otimes H$ that differ only on G's input, then this implies that $G \otimes H$ is not mod-ℓ-respecting either. ◀

If $S = \{G_1, G_2, \ldots\}$, then because of Proposition 2, we can define k on the set S as $\gcd(k(G_1), k(G_2), \ldots)$. For then not only will every transformation in $\langle S \rangle$ be mod-$k(S)$-respecting, but there will exist transformations in $\langle S \rangle$ that are not mod-ℓ-respecting for any $\ell > k(S)$.

We then have that S is mod-k-respecting if and only if k divides $k(S)$, and mod-respecting if and only if S is mod-k-respecting for some $k \geq 2$.

2.3 Alternative Kinds of Generation

We now discuss four alternative notions of what it can mean for a reversible gate set to "generate" a transformation. Besides being interesting in their own right, some of these notions will also be used in the proof of our main classification theorem.

Partial Gates. A *partial reversible gate* is an injective function $H : D \to \{0,1\}^n$, where D is some subset of $\{0,1\}^n$. Such an H is *consistent* with a full reversible gate G if $G(x) = H(x)$ whenever $x \in D$. Also, we say that a reversible gate set S *generates* H if S generates any G with which H is consistent. As an example, COPY is the 2-bit partial reversible gate defined by the following relations:

$$\text{COPY}(00) = 00, \qquad \text{COPY}(10) = 11.$$

If a gate set S can implement the above behavior, using ancilla bits that are returned to their original states by the end, then we say S "generates COPY"; the behavior on inputs 01

and 11 is irrelevant. Note that COPY is consistent with CNOT. One can think of COPY as a bargain-basement CNOT, but one that might be bootstrapped up to a full CNOT with further effort.

Generation With Garbage. Let $D \subseteq \{0,1\}^m$, and $H : D \to \{0,1\}^n$ be some function, which need not be injective or surjective, or even have the same number of input and output bits. Then we say that a reversible gate set S *generates* H *with garbage* if there exists a reversible transformation $G \in \langle S \rangle$, as well as an ancilla string a and a function gar, such that $G(x,a) = (H(x), \text{gar}(x))$ for all $x \in D$. As an example, consider the ordinary 2-bit AND function, from $\{0,1\}^2$ to $\{0,1\}$. Since AND destroys information, clearly no reversible gate can generate it in the usual sense, but many reversible gates can generate AND with garbage: for instance, the Toffoli and Fredkin gates, as we saw in Section 1.1.

Encoded Universality. This is a concept borrowed from quantum computing [5]. In our setting, encoded universality means that there is some way of encoding 0's and 1's by longer strings, such that our gate set can implement any desired transformation on the encoded bits. Note that, while this is a weaker notion of universality than the ability to generate arbitrary permutations of $\{0,1\}^n$, it is stronger than "merely" computational universality, because it still requires a transformation to be performed reversibly, with no garbage left around. Formally, given a reversible gate set S, we say that S *supports encoded universality* if there are k-bit strings $\alpha(0)$ and $\alpha(1)$ such that for every n-bit reversible transformation $F(x_1 \ldots x_n) = y_1 \ldots y_n$, there exists a transformation $G \in \langle S \rangle$ that satisfies

$$G(\alpha(x_1) \ldots \alpha(x_n)) = \alpha(y_1) \ldots \alpha(y_n)$$

for all $x \in \{0,1\}^n$. Also, we say that S *supports affine encoded universality* if this is true for every affine F.

As a well-known example, the Fredkin gate is not universal in the usual sense, because it preserves Hamming weight. But it is easy to see that Fredkin supports encoded universality, using the so-called *dual-rail encoding*, in which every 0 bit is encoded as 01, and every 1 bit is encoded as 10. In Section 4.4, we will show, as a consequence of our classification theorem, that *every* reversible gate set (except for degenerate sets) supports either encoded universality or affine encoded universality.

Loose Generation. Finally, we say that a gate set S *loosely generates* a reversible transformation $F : \{0,1\}^n \to \{0,1\}^n$, if there exists a transformation $G \in \langle S \rangle$, as well as ancilla strings a and b, such that

$$G(x,a) = (F(x), b)$$

for all $x \in \{0,1\}^n$. In other words, G is allowed to change the ancilla bits, so long as they change in a way that is independent of the input x. Under this rule, one could perhaps tell by examining the ancilla bits *that* G was applied, but one could not tell to which input. This suffices for some applications of reversible computing, though not for others.[8]

3 Stating the Classification Theorem

In this section we state our main result, and make a few preliminary remarks about it. First let us define the gates that appear in the classification theorem.

[8] For example, if G were applied to a quantum superposition, then it would still maintain coherence among all the inputs to which it was applied – though perhaps not between those inputs and other inputs in the superposition to which it was *not* applied.

- NOT is the 1-bit gate that maps x to $\overline{x}$.
- NOTNOT, or $\mathrm{NOT}^{\otimes 2}$, is the 2-bit gate that maps x, y to $\overline{x}, \overline{y}$. NOTNOT is a parity-preserving variant of NOT.
- CNOT (Controlled-NOT) is the 2-bit gate that maps x, y to $x, y \oplus x$. CNOT is affine.
- CNOTNOT is the 3-bit gate that maps x, y, z to $x, y \oplus x, z \oplus x$. CNOTNOT is affine and parity-preserving.
- Toffoli (also called Controlled-Controlled-NOT, or CCNOT) is the 3-bit gate that maps x, y, z to $x, y, z \oplus xy$.
- Fredkin (also called Controlled-SWAP, or CSWAP) is the 3-bit gate that maps x, y, z to $x, y \oplus x\,(y \oplus z)\,, z \oplus x\,(y \oplus z)$. In other words, it swaps y with z if $x = 1$, and does nothing if $x = 0$. Fredkin is conservative: it never changes the Hamming weight.
- C_k is a k-bit gate that maps 0^k to 1^k and 1^k to 0^k, and all other k-bit strings to themselves. C_k preserves the Hamming weight mod k. Note that $\mathrm{C}_1 = \mathrm{NOT}$, while C_2 is equivalent to NOTNOT, up to a bit-swap.
- T_k is a k-bit gate (for even k) that maps $x = (x_1, x_2, \ldots, x_k)$ to $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_k)$ if $|x|$ is odd, or to x if $|x|$ is even. A different definition is

  $$\mathrm{T}_k\,(x_1, \ldots, x_k) = (x_1 \oplus b_x, \ldots, x_k \oplus b_x)\,,$$

  where $b_x := x_1 \oplus \cdots \oplus x_k$. This shows that T_k is linear. Indeed, we also have

  $$\mathrm{T}_k\,(x) \cdot \mathrm{T}_k\,(y) \equiv x \cdot y \,(\mathrm{mod}\, 2)\,,$$

  which shows that T_k is orthogonal. Note also that, if $k \equiv 2 \,(\mathrm{mod}\, 4)$, then T_k preserves Hamming weight mod 4: if $|x|$ is even then $|\mathrm{T}_k\,(x)| = |x|$, while if $|x|$ is odd then

  $$|\mathrm{T}_k\,(x)| \equiv k - |x| \equiv 2 - |x| \equiv |x| \,(\mathrm{mod}\, 4)\,.$$

- F_k is a k-bit gate (for even k) that maps x to $\overline{x}$ if $|x|$ is even, or to x if $|x|$ is odd. A different definition is

  $$\mathrm{F}_k\,(x_1 \ldots x_k) = \overline{\mathrm{T}_k\,(x_1 \ldots x_k)} = (x_1 \oplus b_x \oplus 1, \ldots, x_k \oplus b_x \oplus 1)$$

  where b_x is as above. This shows that F_k is affine. Indeed, if k is a multiple of 4, then F_k preserves Hamming weight mod 4: if $|x|$ is odd then $|\mathrm{F}_k\,(x)| = |x|$, while if $|x|$ is even then

  $$|\mathrm{F}_k\,(x)| \equiv k - |x| \equiv |x| \,(\mathrm{mod}\, 4)\,.$$

  Since F_k is equal to T_k in its linear part, F_k is also an isometry.

We can now state the classification theorem.

▶ **Theorem 3** (Main Result). *Every set of reversible gates generates one of the following classes:*

1. *The trivial class (which contains only bit-swaps).*
2. *The class of all transformations (generated by* Toffoli*).*
3. *The class of all conservative transformations (generated by* Fredkin*).*
4. *For each $k \geq 3$, the class of all mod-k-preserving transformations (generated by C_k).*
5. *The class of all affine transformations (generated by* CNOT*).*
6. *The class of all parity-preserving affine transformations (generated by* CNOTNOT*).*
7. *The class of all mod-4-preserving affine transformations (generated by F_4).*

8. *The class of all orthogonal linear transformations (generated by* T_4*).*
9. *The class of all mod-*4*-preserving orthogonal linear transformations (generated by* T_6*).*
10. *Classes 1, 3, 7, 8, or 9 augmented by a* NOTNOT *gate (note: 7 and 8 become equivalent this way).*
11. *Classes 1, 3, 6, 7, 8, or 9 augmented by a* NOT *gate (note: 7 and 8 become equivalent this way).*

Furthermore, all the above classes are distinct except when noted otherwise, and they fit together in the lattice diagram shown in Figure 3.[9]

Let us make some comments about the structure of the lattice. The lattice has a countably infinite number of classes, with the one infinite part given by the mod-k-preserving classes. The mod-k-preserving classes are partially ordered by divisibility, which means, for example, that the lattice is not planar.[10] While there are infinite descending chains in the lattice, there is no infinite ascending chain. This means that, if we start from some reversible gate class and then add new gates that extend its power, we must terminate after finitely many steps with the class of all reversible transformations.

In the full version [4], we prove that if we allow loose generation, then the only change to Theorem 3 is that every class $\mathcal{C}$ containing a NOTNOT gate collapses with $\mathcal{C} + \mathrm{NOT}$.

4 Consequences of the Classification

To illustrate the power of the classification theorem, in this section we use it to prove four general implications for reversible computation. While these implications are easy to prove with the classification in hand, we do not know how to prove any of them without it.

4.1 Nature of the Classes

Here is one immediate (though already non-obvious) corollary of Theorem 3.

▶ **Corollary 4.** *Every reversible gate class* $\mathcal{C}$ *is finitely generated: that is, there exists a finite set* S *such that* $\mathcal{C} = \langle S \rangle$*.*

Indeed, we have something stronger.

▶ **Corollary 5.** *Every reversible gate class* $\mathcal{C}$ *is generated by a single gate* $G \in \mathcal{C}$*.*

Proof. This is immediate for all the classes listed in Theorem 3, except the ones involving NOT or NOTNOT gates. For classes of the form $\mathcal{C} = \langle G, \mathrm{NOT} \rangle$ or $\mathcal{C} = \langle G, \mathrm{NOTNOT} \rangle$, we just need a single gate G' that is clearly generated by $\mathcal{C}$, and clearly *not* generated by a smaller class. We can then appeal to Theorem 3 to assert that G' *must* generate $\mathcal{C}$. For each of the relevant G's – namely, Fredkin, CNOTNOT, F_4, and T_6 – one such G' is the tensor product, $G \otimes \mathrm{NOT}$ or $G \otimes \mathrm{NOTNOT}$. ◀

[9] Let us mention that Fredkin + NOTNOT generates the class of all parity-preserving transformations, while Fredkin + NOT generates the class of all parity-respecting transformations. We could have listed the parity-preserving transformations as a special case of the mod-k-preserving transformations: namely, the case $k = 2$. If we had done so, though, we would have had to include the caveat that C_k only generates all mod-k-preserving transformations when $k \geq 3$ (when $k = 2$, we also need Fredkin in the generating set). And in any case, the parity-respecting class would still need to be listed separately.

[10] For consider the graph with the integers 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 15, 18, 20, 21, 24, and 28 as its vertices, and with an edge between each pair whose ratio is a prime. One can check that this graph contains $K_{3,3}$ as a minor.

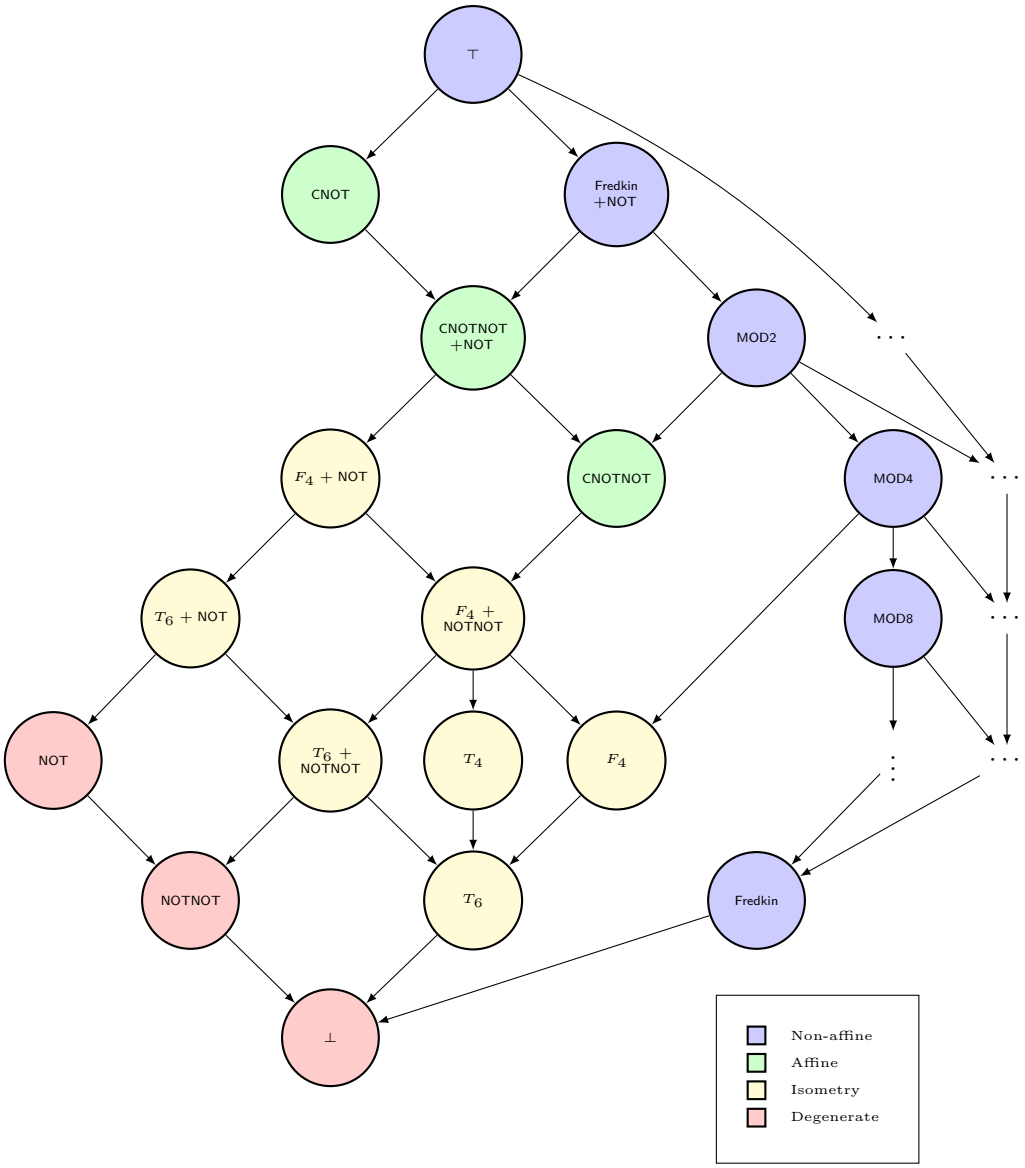


Figure 3 The inclusion lattice of reversible gate classes.

We also wish to point out a non-obvious symmetry property that follows from the classification theorem. Given an n-bit reversible transformation F, let F^*, or the *dual* of F, be $F^*(x_1 \dots x_n) := \overline{F(\overline{x}_1 \dots \overline{x}_n)}$. The dual can be thought of as F with the roles of 0 and 1 interchanged: for example, Toffoli$^*(xyz)$ flips z if and only if $x = y = 0$. Also, call a gate F *self-dual* if $F^* = F$, and call a reversible gate class $\mathcal{C}$ *dual-closed* if $F^* \in \mathcal{C}$ whenever $F \in \mathcal{C}$. Then:

▶ **Corollary 6.** *Every reversible gate class $\mathcal{C}$ is dual-closed.*

Proof. This is obvious for all the classes listed in Theorem 3 that include a NOT or NOTNOT gate. For the others, we simply need to consider the classes one by one: the notions of "conservative," "mod-k-respecting," and "mod-k-preserving" are manifestly the same after we interchange 0 and 1. This is less manifest for the notion of "orthogonal," but one can check that T_k and F_k are self-dual for all even k. ◀

4.2 Linear-Time Algorithm

If one wanted, one could interpret this entire paper as addressing a straightforward *algorithms* problem: namely, the RevGen problem defined in Section 1.4, where we are given as input a set of reversible gates $G_1, \ldots, G_K$, as well as a target reversible transformation H, and we want to know whether the G_i's generate H. From that perspective, our contribution is to reduce the known upper bound on the complexity of RevGen: from recursively-enumerable (!), or triply-exponential time if we use Jeřábek's recent clone/coclone duality for reversible gates [15], all the way down to linear time.

▶ **Theorem 7.** *There is a linear-time algorithm for* RevGen.

Proof. It suffices to give a linear-time algorithm that takes as input the truth table of a single reversible transformation $G : \{0,1\}^n \to \{0,1\}^n$, and that decides which class it generates. For we can then compute $\langle G_1, \ldots, G_K \rangle$ by taking the least upper bound of $\langle G_1 \rangle, \ldots, \langle G_K \rangle$, and can also solve the membership problem by checking whether

$$\langle G_1, \ldots, G_K \rangle = \langle G_1, \ldots, G_K, H \rangle .$$

The algorithm is as follows: first, make a single pass through G's truth table, in order to answer the following two questions.

- Is G affine, and if so, what is its matrix representation, $G(x) = Ax \oplus b$?
- What is $W(G) := \{|G(x)| - |x| : x \in \{0,1\}^n\}$?

In any reasonable RAM model, both questions can easily be answered in $O(n2^n)$ time, which is the number of bits in G's truth table.

If G is non-affine, then Theorem 3 implies that we can determine $\langle G \rangle$ from $W(G)$ alone. If G is affine, then Theorem 3 implies we can determine $\langle G \rangle$ from (A, b) alone, though it is also convenient to use $W(G)$. We need to take the gcd of the numbers in $W(G)$, check whether A is orthogonal, etc., but the time needed for these operations is only $\text{poly}(n)$, which is negligible compared to the input size of $n2^n$. ◀

We have implemented the algorithm described in Theorem 7, and Java code is available for download [25].

4.3 Compression of Reversible Circuits

We now state a "complexity-theoretic" consequence of Theorem 3.

▶ **Theorem 8.** *Let R be a reversible circuit, over any gate set S, that maps $\{0,1\}^n$ to $\{0,1\}^n$, using an unlimited number of gates and ancilla bits. Then there is another reversible circuit, over the same gate set S, that applies the same transformation as R does, and that uses only $2^n \, \text{poly}(n)$ gates and $O(1)$ ancilla bits.*[11]

Proof. If S is one of the gate sets listed in Theorem 3, then this follows immediately by examining the reversible circuit constructions in Section 7, for each class in the classification. Building, in relevant parts, on results by others [26, 7], we will take care in Section 7 to ensure that each non-affine circuit construction uses at most $2^n \, \text{poly}(n)$ gates and $O(1)$ ancilla bits, while each affine construction uses at most $O(n^2)$ gates and $O(1)$ ancilla bits (most actually use no ancilla bits).

[11] Here the big-O's suppress constant factors that depend on the gate set in question.

Now suppose S is *not* one of the sets listed in Theorem 3, but some other set that generates one of the listed classes. So for example, suppose $\langle S \rangle = \langle \text{Fredkin}, \text{NOT} \rangle$. Even then, we know that S *generates* Fredkin and NOT, and the number of gates and ancillas needed to do so is just some constant, independent of n. Furthermore, each time we need a Fredkin or NOT, we can reuse the same ancilla bits, by the assumption that those bits are returned to their original states. So we can simply simulate the appropriate circuit construction from Section 7, using only a constant factor more gates and $O(1)$ more ancilla bits than the original construction. ◀

As we said in Section 1.4, without the classification theorem, it is not obvious how to prove *any upper bound whatsoever* on the number of gates or ancillas, for arbitrary gate sets S. Of course, any circuit that uses T gates also uses at most $O(T)$ ancillas; and conversely, any circuit that uses M ancillas needs at most $\left(2^{n+M}\right)!$ gates, for counting reasons. But the best upper bounds on either quantity that follow from clone theory and the ideal membership problem appear to have the form $\exp(\exp(\exp(\exp(n))))$.

A constant number of ancilla bits *is* sometimes needed, and not only for the trivial reasons that our gates might act on more than n bits, or only (e.g.) be able to map 0^n to 0^n if no ancillas are available.

▶ **Proposition 9** (Toffoli [29]). *If no ancillas are allowed, then there exist reversible transformations of* $\{0,1\}^n$ *that cannot be generated by any sequence of reversible gates on* $n-1$ *bits or fewer.*

Proof. For all $k \geq 1$, any $(n-k)$-bit gate induces an even permutation of $\{0,1\}^n$ – since each cycle is repeated 2^k times, once for every setting of the k bits on which the gate doesn't act. But there are also odd permutations of $\{0,1\}^n$. ◀

It is also easy to show, using a Shannon counting argument, that there exist n-bit reversible transformations that require $\Omega(2^n)$ gates to implement, and n-bit affine transformations that require $\Omega\left(n^2/\log n\right)$ gates. Thus the bounds in Theorem 8 on the number of gates T are, for each class, off from the optimal bounds only by $\text{polylog}\, T$ factors.

4.4 Encoded Universality

If we only care about which Boolean functions $f : \{0,1\}^n \to \{0,1\}$ can be computed, and are completely uninterested in what garbage is output along with f, then it is not hard to see that all reversible gate sets fall into three classes. Namely, non-affine gate sets (such as Toffoli and Fredkin) can compute all Boolean functions;[12] non-degenerate affine gate sets (such as CNOT and CNOTNOT) can compute all affine functions; and degenerate gate sets (such as NOT and NOTNOT) can compute only 1-bit functions. However, the classification theorem lets us make a more interesting statement. Recall the notion of *encoded universality* from Section 2.3, which demands that every reversible transformation (or every affine transformation) be implementable without garbage, once 0 and 1 are "encoded" by longer strings $\alpha(0)$ and $\alpha(1)$ respectively.

▶ **Theorem 10.** *Besides the trivial,* NOT*, and* NOTNOT *classes, every reversible gate class supports encoded universality if non-affine, or affine encoded universality if affine.*

[12] This was proven by Lloyd [21], as well as by Kerntopf et al. [16] and De Vos and Storme [30]; we include a proof for completeness in the full version of this paper [4].

Proof. For $\langle \text{Fredkin} \rangle$, and for all the non-affine classes above $\langle \text{Fredkin} \rangle$, we use the so-called "dual-rail encoding," where 0 is encoded by 01 and 1 is encoded by 10. Given three encoded bits, $x\overline{x}y\overline{y}z\overline{z}$, we can simulate a Fredkin gate by applying one Fredkin to xyz and another to $x\overline{y}\overline{z}$, and can also simulate a CNOT by applying a Fredkin to $xy\overline{y}$. But Fredkin + CNOT generates everything.

The dual-rail encoding also works for simulating all affine transformations using an F_4 gate. For note that

$$\begin{aligned} \mathrm{F}_4\left(xy\overline{y}1\right) &= \left(1, \overline{x \oplus y}, x \oplus y, x\right) \\ &= \left(x, x \oplus y, \overline{x \oplus y}, 1\right), \end{aligned}$$

where we used that we can permute bits for free. So given two encoded bits, $x\overline{x}y\overline{y}$, we can simulate a CNOT from x to y by applying F_4 to x, y, $\overline{y}$, and one ancilla bit initialized to 1.

For $\langle \text{CNOTNOT} \rangle$, we use a repetition encoding, where 0 is encoded by 00 and 1 is encoded by 11. Given two encoded bits, $xxyy$, we can simulate a CNOT from x to y by applying a CNOTNOT from either copy of x to both copies of y. This lets us perform all affine transformations on the encoded subspace.

The repetition encoding also works for $\langle \mathrm{T}_4 \rangle$. For notice that

$$\begin{aligned} \mathrm{T}_4\left(xyy0\right) &= \left(0, x \oplus y, x \oplus y, x\right) \\ &= \left(x, x \oplus y, x \oplus y, 0\right). \end{aligned}$$

Thus, to simulate a CNOT from x to y, we use one copy of x, both copies of y, and one ancilla bit initialized to 0.

Finally, for $\langle \mathrm{T}_6 \rangle$, we encode 0 by 0011 and 1 by 1100. Notice that

$$\begin{aligned} \mathrm{T}_6\left(xyy\overline{y}\overline{y}0\right) &= \left(0, x \oplus y, x \oplus y, \overline{x \oplus y}, \overline{x \oplus y}, x\right) \\ &= \left(x, x \oplus y, x \oplus y, \overline{x \oplus y}, \overline{x \oplus y}, 0\right). \end{aligned}$$

So given two encoded bits, $xx\overline{x}\overline{x}yy\overline{y}\overline{y}$, we can simulate a CNOT from x to y by using one copy of x, all four copies of y and $\overline{y}$, and one ancilla bit initialized to 0. ◀

In the proof of Theorem 10, notice that, every time we simulated Fredkin (xyz) or CNOT (xy), we had to examine only a single bit in the encoding of the control bit x. Thus, Theorem 10 actually yields a stronger consequence: that given an ordinary, unencoded input string $x_1 \ldots x_n$, we can use any non-degenerate reversible gate first to *translate* x into its encoded version $\alpha\left(x_1\right) \ldots \alpha\left(x_n\right)$, and then to perform arbitrary transformations or affine transformations on the encoding.

5 Structure of the Proof

The proof of Theorem 3 naturally divides into four components. First, we need to verify that all the gates mentioned in the theorem really do satisfy the invariants that they are claimed to satisfy – and as a consequence, that any reversible transformation they generate also satisfies the invariants. This is completely routine.

Second, we need to verify that all pairs of classes that Theorem 3 says are distinct, *are* distinct. We handle this in Theorem 11 below (there are only a few non-obvious cases).

Third, we need to verify that the "gate definition" of each class coincides with its "invariant definition"– i.e., that each gate really does generate all reversible transformations that satisfy its associated invariant. For example, we need to show that Fredkin generates all conservative

transformations, that C_k generates all transformations that preserve Hamming weight mod k, and that T_4 generates all orthogonal linear transformations. Many of these results are already known, but for completeness, we prove all of them in Section 7, by giving explicit constructions of reversible circuits.[13]

Finally, we need to show that there are no *additional* reversible gate classes, besides the ones listed in Theorem 3. This is by far the most interesting part, but it appears only in the full version of the paper [4] due to its length. Nevertheless, the organization of the complete proof is as follows:

- In Section 6, we collect numerous results about what reversible transformations can and cannot do to Hamming weights mod k and inner products mod k, in both the affine and the non-affine cases; these results are then drawn on in the rest of the paper. (Some of them are even used for the circuit constructions in Section 7.)
- In the full version, we complete the classification of all non-affine gate sets. We show that the only classes that contain a Fredkin gate (equivalently, the classes above $\langle \text{Fredkin} \rangle$ in the lattice) are $\langle \text{Fredkin} \rangle$ itself, $\langle \text{Fredkin}, \text{NOTNOT} \rangle$, $\langle \text{Fredkin}, \text{NOT} \rangle$, $\langle C_k \rangle$ for $k \geq 3$, and $\langle \text{Toffoli} \rangle$. Next, we show that every nontrivial conservative gate generates Fredkin. We then build on that result to show that every non-affine gate set generates Fredkin.
- The complete classification of all affine gate sets is also contained in the full version. For simplicity, we start with *linear* gate sets only, and show that every nontrivial mod-4-preserving linear gate generates T_6, and that every nontrivial, *non*-mod-4-preserving orthogonal gate generates T_4. Next, we show that every non-orthogonal linear gate generates CNOTNOT. Then, we show that every non-parity-preserving linear gate generates CNOT. Since CNOT generates all linear transformations, this completes the classification of linear gate sets. Finally, we "put back the affine part," showing that it can lead to only 8 additional classes besides the linear classes $\langle \varnothing \rangle$, $\langle T_6 \rangle$, $\langle T_4 \rangle$, $\langle \text{CNOTNOT} \rangle$, and $\langle \text{CNOT} \rangle$.

▶ **Theorem 11.** *All pairs of classes asserted to be distinct by Theorem 3, are distinct.*

Proof. In each case, one just needs to observe that the gate that generates a given class A, satisfies some invariant violated by the gate that generates another class B. (Here we are using the "gate definitions" of the classes, which will be proven equivalent to the invariant definitions in Section 7.) So for example, $\langle \text{Fredkin} \rangle$ cannot contain CNOT because Fredkin is conservative; conversely, $\langle \text{CNOT} \rangle$ cannot contain Fredkin because CNOT is affine.

The only tricky classes are those involving NOT and NOTNOT gates: indeed, these classes *do* sometimes coincide, as noted in Theorem 3. However, in all cases where the classes are distinct, their distinctness is witnessed by the following invariants:

- $\langle \text{Fredkin}, \text{NOT} \rangle$ and $\langle \text{Fredkin}, \text{NOTNOT} \rangle$ are conservative in their linear part.
- $\langle \text{CNOTNOT}, \text{NOT} \rangle$ is parity-preserving in its linear part.
- $\langle F_4, \text{NOT} \rangle = \langle T_4, \text{NOT} \rangle$ and $\langle F_4, \text{NOTNOT} \rangle = \langle T_4, \text{NOTNOT} \rangle$ are orthogonal in their linear part (isometries).
- $\langle T_6, \text{NOT} \rangle$ and $\langle T_6, \text{NOTNOT} \rangle$ are orthogonal and mod-4-preserving in their linear part.

[13] The upshot of the Galois connection for clones [15] is that, if we could prove that a list of invariants for a given gate set S was the *complete* list of invariants satisfied by S, then this second part of the proof would be unnecessary: it would follow automatically that S generates all reversible transformations that satisfy the invariants. But this raises the question: how do we prove that a list of invariants for S is complete? In each case, the easiest way we could find to do this, was just by explicitly describing circuits of S-gates to generate all transformations that satisfy the stated invariants.

As a final remark, even if a reversible transformation is implemented with the help of ancilla bits, as long as the ancilla bits start and end in the same state $a_1 \dots a_k$, they have no effect on any of the invariants discussed above, and for that reason are irrelevant. ◀

6 Hamming Weights and Inner Products

The purpose of this section is to collect various mathematical results about what a reversible transformation $G : \{0,1\}^n \to \{0,1\}^n$ can and cannot do to the Hamming weight of its input, or to the inner product of two inputs. That is, we study the possible relationships that can hold between $|x|$ and $|G(x)|$, or between $x \cdot y$ and $G(x) \cdot G(y)$ (especially modulo various positive integers k). Not only are these results used heavily in the rest of the classification, but some of them might be of independent interest.

6.1 Ruling Out Mod-Shifters

Call a reversible transformation a *mod-shifter* if it always shifts the Hamming weight mod k of its input string by some fixed, nonzero amount. When $k = 2$, clearly mod-shifters exist: indeed, the humble NOT gate satisfies $|\text{NOT}(x)| \equiv |x| + 1 \pmod 2$ for all $x \in \{0,1\}$, and likewise for any other parity-flipping gate. However, we now show that $k = 2$ is the *only* possibility: mod-shifters do not exist for any larger k.

▶ **Theorem 12.** *There are no mod-shifters for $k \geq 3$. In other words: let G be a reversible transformation on n-bit strings, and suppose*

$$|G(x)| \equiv |x| + j \pmod k$$

for all $x \in \{0,1\}^n$. Then either $j = 0$ or $k = 2$.

Proof. Suppose the above equation holds for all x. Then introducing a new complex variable z, we have

$$z^{|G(x)|} \equiv z^{|x|+j} \left(\text{mod } \left(z^k - 1\right)\right)$$

(since working mod $z^k - 1$ is equivalent to setting $z^k = 1$). Since the above is true for all x,

$$\sum_{x \in \{0,1\}^n} z^{|G(x)|} \equiv \sum_{x \in \{0,1\}^n} z^{|x|} z^j \left(\text{mod } \left(z^k - 1\right)\right). \tag{1}$$

By reversibility, we have

$$\sum_{x \in \{0,1\}^n} z^{|G(x)|} = \sum_{x \in \{0,1\}^n} z^{|x|} = (z+1)^n.$$

Therefore equation (1) simplifies to

$$(z+1)^n \left(z^j - 1\right) \equiv 0 \left(\text{mod } \left(z^k - 1\right)\right).$$

Now, since $z^k - 1$ has no repeated roots, it can divide $(z+1)^n \left(z^j - 1\right)$ only if it divides $(z+1)\left(z^j - 1\right)$. For this we need either $j = 0$, causing $z^j - 1 = 0$, or else $j = k - 1$ (from degree considerations). But it is easily checked that the equality

$$z^k - 1 = (z+1)\left(z^{k-1} - 1\right)$$

holds only if $k = 2$. ◀

6.2 Inner Products Mod k

We have seen that there exist *orthogonal* gates (such as the T_k gates), which preserve inner products mod 2. In this section, we first show that no reversible gate that changes Hamming weights can preserve inner products mod k for any $k \geq 3$. We then observe that, if a reversible gate is orthogonal, then it must be linear, and we give necessary and conditions for orthogonality.

▶ **Theorem 13.** *Let G be a non-conservative n-bit reversible gate, and suppose*

$$G(x) \cdot G(y) \equiv x \cdot y \pmod{k}$$

for all $x, y \in \{0,1\}^n$. Then $k = 2$.

Proof. As in the proof of Theorem 12, we promote the congruence to a congruence over complex polynomials:

$$z^{G(x)\cdot G(y)} \equiv z^{x\cdot y} \pmod{z^k - 1}$$

Fix a string $x \in \{0,1\}^n$ such that $|G(x)| > |x|$, which must exist because G is non-conservative. Then sum the congruence over all y:

$$\sum_{y\in\{0,1\}^n} z^{G(x)\cdot G(y)} \equiv \sum_{y\in\{0,1\}^n} z^{x\cdot y} \pmod{z^k - 1}.$$

The summation on the right simplifies as follows.

$$\begin{aligned}\sum_{y\in\{0,1\}^n} z^{x\cdot y} &= \sum_{y\in\{0,1\}^n} \prod_{i=1}^{n} z^{x_i y_i} = \prod_{i=1}^{n} \sum_{y_i\in\{0,1\}} z^{x_i y_i} = \prod_{i=1}^{n} (1 + z^{x_i}) = (1+z)^{|x|}\, 2^{n-|x|}, \\ &= \left(\frac{1+z}{2}\right)^{|x|} 2^n.\end{aligned}$$

Similarly,

$$\sum_{y\in\{0,1\}^n} z^{G(x)\cdot G(y)} = \left(\frac{1+z}{2}\right)^{|G(x)|} 2^n,$$

since summing over all y is the same as summing over all $G(y)$. So we have

$$\begin{aligned}\left(\frac{1+z}{2}\right)^{|G(x)|} 2^n &\equiv \left(\frac{1+z}{2}\right)^{|x|} 2^n \pmod{z^k - 1}, \\ 0 &\equiv (1+z)^{|x|} 2^{-|G(x)|} \left(2^{|G(x)|-|x|} - (1+z)^{|G(x)|-|x|}\right) \pmod{z^k - 1},\end{aligned}$$

or equivalently, letting

$$p(x) := 2^{|G(x)|-|x|} - (1+z)^{|G(x)|-|x|},$$

we find that $z^k - 1$ divides $(1+z)^{|x|} p(x)$ as a polynomial. Now, the roots of $z^k - 1$ lie on the unit circle centered at 0. Meanwhile, the roots of $p(x)$ lie on the circle in the complex plane of radius 2, centered at -1. The only point of intersection of these two circles is $z = 1$, so that is the only root of $z^k - 1$ that can be covered by $p(x)$. On the other hand, clearly $z = -1$ is the only root of $(1+z)^{|x|}$. Hence, the only roots of $z^k - 1$ are 1 and -1, so we conclude that $k = 2$. ◀

We now study reversible transformations that preserve inner products mod 2.

▶ **Lemma 14.** *Every orthogonal gate G is linear.*

Proof. Suppose

$$G(x) \cdot G(y) \equiv x \cdot y \pmod 2.$$

Then for all x, y, z,

$$\begin{aligned} G(x \oplus y) \cdot G(z) &\equiv (x \oplus y) \cdot z \\ &\equiv x \cdot z + y \cdot z \\ &\equiv G(x) \cdot G(z) + G(y) \cdot G(z) \\ &\equiv (G(x) \oplus G(y)) \cdot G(z) \pmod 2. \end{aligned}$$

But if the above holds for all possible z, then

$$G(x \oplus y) \equiv G(x) \oplus G(y) \pmod 2. \quad \blacktriangleleft$$

Theorem 13 and Lemma 14 have the following corollary.

▶ **Corollary 15.** *Let G be any non-conservative, nonlinear gate. Then for all $k \geq 2$, there exist inputs x, y such that*

$$G(x) \cdot G(y) \not\equiv x \cdot y \pmod k.$$

Also:

▶ **Lemma 16.** *A linear transformation $G(x) = Ax$ is orthogonal if and only if $A^T A$ is the identity: that is, if A's column vectors satisfy $|v_i| \equiv 1 \pmod 2$ for all i and $v_i \cdot v_j \equiv 0 \pmod 2$ for all $i \neq j$.*

Proof. This is just the standard characterization of orthogonal matrices; that we are working over $\mathbb{F}_2$ is irrelevant. First, if G preserves inner products mod 2 then for all $i \neq j$,

$$\begin{aligned} 1 &\equiv e_i \cdot e_i \equiv (Ae_i) \cdot (Ae_i) \equiv |v_i| \pmod 2, \\ 0 &\equiv e_i \cdot e_j \equiv (Ae_i) \cdot (Ae_j) \equiv v_i \cdot v_j \pmod 2. \end{aligned}$$

Second, if G satisfies the conditions then

$$Ax \cdot Ay \equiv (Ax)^T Ay \equiv x^T (A^T A) y \equiv x^T y \equiv x \cdot y \pmod 2. \quad \blacktriangleleft$$

6.3 Why Mod 2 and Mod 4 Are Special

Recall that $\wedge$ denotes bitwise AND. We first need an "inclusion/exclusion formula" for the Hamming weight of a bitwise sum of strings.

▶ **Lemma 17.** *For all $v_1, \ldots, v_t \in \{0,1\}^n$, we have*

$$|v_1 \oplus \cdots \oplus v_t| = \sum_{\emptyset \subset S \subseteq [t]} (-2)^{|S|-1} \left| \bigwedge_{i \in S} v_i \right|.$$

Proof. It suffices to prove the lemma for $n = 1$, since in the general case we are just summing over all $i \in [n]$. Thus, assume without loss of generality that $v_1 = \cdots = v_t = 1$. Our problem then reduces to proving the following identity:

$$\sum_{i=1}^{t}(-2)^{i-1}\binom{t}{i} = \begin{cases} 0 & \text{if } t \text{ is even} \\ 1 & \text{if } t \text{ is odd,} \end{cases}$$

which follows straightforwardly from the binomial theorem. ◀

▶ **Lemma 18.** *No nontrivial affine gate G is conservative.*

Proof. Let $G(x) = Ax \oplus b$; then $|G(0^n)| = |0^n| = 0$ implies $b = 0^n$. Likewise, $|G(e_i)| = |e_i| = 1$ for all i implies that A is a permutation matrix. But then G is trivial. ◀

▶ **Theorem 19.** *If G is a nontrivial linear gate that preserves Hamming weight mod k, then either $k = 2$ or $k = 4$.*

Proof. For all x, y, we have

$$\begin{aligned} |x| + |y| - 2(x \cdot y) &\equiv |x \oplus y| \\ &\equiv |G(x \oplus y)| \\ &\equiv |G(x) \oplus G(y)| \\ &\equiv |G(x)| + |G(y)| - 2(G(x) \cdot G(y)) \\ &\equiv |x| + |y| - 2(G(x) \cdot G(y)) \pmod{k}, \end{aligned}$$

where the first and fourth lines used Lemma 17, the second and fifth lines used that G is mod-k-preserving, and the third line used linearity. Hence

$$2(x \cdot y) \equiv 2(G(x) \cdot G(y)) \pmod{k}. \tag{2}$$

If k is odd, then equation (2) implies

$$x \cdot y \equiv G(x) \cdot G(y) \pmod{k}.$$

But since G is nontrivial and linear, Lemma 18 says that G is non-conservative. So by Theorem 13, the above equation cannot be satisfied for any odd $k \geq 3$. Likewise, if k is even, then (2) implies

$$x \cdot y \equiv G(x) \cdot G(y) \left(\mod \frac{k}{2}\right).$$

Again by Theorem 13, the above can be satisfied only if $k = 2$ or $k = 4$. ◀

▶ **Theorem 20.** *Let $\{o_i\}_{i=1}^n$ be an orthonormal basis over $\mathbb{F}_2$. An affine transformation $F(x) = Ax \oplus b$ is mod-4-preserving if and only if $|b| \equiv 0 \pmod 4$, and the vectors $v_i := Ao_i$ satisfy $|v_i| + 2(v_i \cdot b) \equiv |o_i| \pmod 4$ for all i and $v_i \cdot v_j \equiv 0 \pmod 2$ for all $i \neq j$.*

Proof. First, if F is mod-4-preserving, then

$$0 \equiv |F(0^n)| \equiv |A0^n \oplus b| \equiv |b| \pmod 4,$$

and hence

$$|o_i| \equiv |F(o_i)| \equiv |Ao_i \oplus b| \equiv |v_i \oplus b| \equiv |v_i| + |b| - 2(v_i \cdot b) \equiv |v_i| + 2(v_i \cdot b) \pmod 4$$

for all i, and hence

$$\begin{aligned}|o_i + o_j| &\equiv |F(o_i \oplus o_j)| \\ &\equiv |v_i \oplus v_j \oplus b| \\ &\equiv |v_i| + |v_j| + |b| - 2(v_i \cdot v_j) - 2(v_i \cdot b) - 2(v_j \cdot b) + 4|v_i \wedge v_j \wedge b| \\ &\equiv |v_i| + |v_j| + 2(v_i \cdot v_j) + 2(v_i \cdot b) + 2(v_j \cdot b) \pmod 4 \\ &\equiv |o_i| + |o_j| + 2(v_i \cdot v_j) \pmod 4\end{aligned}$$

for all $i \neq j$, from which we conclude that $v_i \cdot v_j \equiv 0 \pmod 2$.

Second, if F satisfies the conditions, then for any $x = \sum_{i \in S} o_i$, we have

$$\begin{aligned}|F(x)| &= \left| b \oplus \sum_{i \in S} v_i \right| \\ &= |b| + \sum_{i \in S} |v_i| - 2\sum_{i \in S} (b \cdot v_i) - 2 \sum_{i \in S \,<\, j \in S} (v_i \cdot v_j) + 4(\cdots) \\ &\equiv \sum_{i \in S} |v_i| - 2(b \cdot v_i) \\ &\equiv \sum_{i \in S} |o_i| \pmod 4,\end{aligned}$$

where the second line follows from Lemma 17. Furthermore, we have that

$$|x| = \left| \sum_{i \in S} o_i \right| = \sum_{i \in S} |o_i| - 2 \sum_{i \in S < j \in S} (o_i \cdot o_j) + 4(\ldots) \equiv \sum_{i \in S} |o_i| \pmod 4,$$

where the last equality follows from the fact that $\{o_i\}_{i=1}^n$ is an orthonormal basis. Therefore, we conclude that $|F(x)| \equiv |x| \pmod 4$. ◀

We note two corollaries of Theorem 20 for later use.

▶ **Corollary 21.** *Any linear transformation $A \in \mathbb{F}_2^{n \times n}$ that preserves Hamming weight mod 4 is also orthogonal.*

▶ **Corollary 22.** *An orthogonal transformation $A \in \mathbb{F}_2^{n \times n}$ preserves Hamming weight mod 4 if and only if all of its columns have Hamming weight 1 mod 4.*

7 Reversible Circuit Constructions

In this section, we show that all the classes of reversible transformations listed in Theorem 3, are indeed generated by the gates that we claimed, by giving explicit synthesis procedures. In order to justify Theorem 8, we also verify that in each case, only $O(1)$ ancilla bits are needed, even though this constraint makes some of the constructions more complicated than otherwise.

Many of our constructions – those for Toffoli and CNOT, for example – have appeared in various forms in the reversible computing literature, and are included here only for completeness. Others – those for C_k and F_4, for example – are new as far as we know, but not hard.

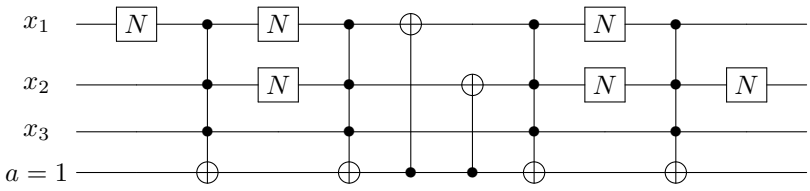


Figure 4 Circuit for the transposition $\sigma_{011,101}$ after simplification.

7.1 Non-Affine Circuits

We start with the non-affine classes: $\langle \text{Toffoli} \rangle$, $\langle \text{Fredkin} \rangle$, $\langle \text{Fredkin}, \text{C}_k \rangle$, and $\langle \text{Fredkin}, \text{NOT} \rangle$.

▶ **Theorem 23** (variants in [29, 26]). Toffoli *generates all reversible transformations on n bits, using only 2 ancilla bits.*[14]

Proof. Any reversible transformation $F : \{0,1\}^n \to \{0,1\}^n$ is a permutation of n-bit strings, and any permutation can be written as a product of transpositions. So it suffices to show how to use Toffoli gates to implement an arbitrary transposition $\sigma_{y,z}$: that is, a mapping that sends $y = y_1 \dots y_n$ to $z = z_1 \dots z_n$ and z to y, and all other n-bit strings to themselves.

Given any n-bit string w, let us define w-CNOT to be the $(n+1)$-bit gate that flips its last bit if its first n bits are equal to w, and that does nothing otherwise. (Thus, the Toffoli gate is 11-CNOT, while CNOT itself is 1-CNOT.) Given y-CNOT and z-CNOT gates, we can implement the transposition $\sigma_{y,z}$ as follows on input x:

1. Initialize an ancilla bit, $a = 1$.
2. Apply y-CNOT (x, a).
3. Apply z-CNOT (x, a).
4. Apply NOT gates to all x_i's such that $y_i \neq z_i$.
5. For each i such that $y_i \neq z_i$, apply CNOT (a, x_i).
6. Apply z-CNOT (x, a).
7. Apply y-CNOT (x, a).

Thus, all that remains is to implement w-CNOT using Toffoli. Observe that we can simulate any w-CNOT using 1^n-CNOT by negating certain input bits (namely, those for which $w_i = 0$) before and after we apply the 1^n-CNOT. An example of the transposition $\sigma_{011,101}$ is given in Figure 4.

So it suffices to implement 1^n-CNOT, with control bits $x_1 \dots x_n$ and target bit y. The base case is $n = 2$, which we implement directly using Toffoli. For $n \geq 3$, we do the following.

- Let a be an ancilla.
- Apply $1^{\lceil n/2 \rceil}$-CNOT $\left(x_1 \dots x_{\lceil n/2 \rceil}, a\right)$.
- Apply $1^{\lfloor n/2 \rfloor + 1}$-CNOT $\left(x_{\lceil n/2 \rceil + 1} \dots x_n, a, y\right)$.
- Apply $1^{\lceil n/2 \rceil}$-CNOT $\left(x_1 \dots x_{\lceil n/2 \rceil}, a\right)$.
- Apply $1^{\lfloor n/2 \rfloor + 1}$-CNOT $\left(x_{\lceil n/2 \rceil + 1} \dots x_n, a, y\right)$.

The crucial point is that this construction works whether the ancilla is initially 0 or 1. In other words, we can use *any* bit which is not one of the inputs, instead of a new ancilla. For instance, we can have one bit dedicated for use in 1^n-CNOT gates, which we use in the

[14] Notice that we need at least 2 so that we can generate CNOT and NOT using Toffoli.

recursive applications of $1^{\lceil n/2 \rceil}$-CNOT and $1^{\lfloor n/2 \rfloor + 1}$-CNOT, and the recursive applications within them, and so on.[15]

Carefully inspecting the above proof shows that $O\left(n^2 2^n\right)$ gates and 2 ancilla bits suffice to generate any transformation. ◀

The particular construction above was inspired by a result of Ben-Or and Cleve [7], in which they compute algebraic formulas in a straight-line computation model using a constant number of registers. We note that Toffoli [29] proved a version of Theorem 23, but with $O(n)$ ancilla bits rather than $O(1)$. More recently, Shende et al. [26] gave a slightly more complicated construction that uses only 1 ancilla bit (assuming that we have CNOT and NOT gates in addition to Toffoli gates), and that also gives explicit bounds on the number of Toffoli gates required based on the number of fixed points of the permutation. Recall that at least 1 ancilla bit is needed by Proposition 9.

Next, let CCSWAP, or Controlled-Controlled-SWAP, be the 4-bit gate that swaps its last two bits if its first two bits are both 1, and otherwise does nothing.

▶ **Proposition 24.** Fredkin *generates* CCSWAP.

Proof. Let a be an ancilla bit initialized to 0. We implement CCSWAP (x, y, z, w) by applying Fredkin (x, y, a), then Fredkin (a, z, w), then again Fredkin (x, y, a). ◀

We can now prove an analogue of Theorem 23 for conservative transformations.

▶ **Theorem 25.** Fredkin *generates all conservative transformations on n bits, using only* 5 *ancilla bits.*

Proof. In this proof, we will use the *dual-rail representation*, in which 0 is encoded as 01 and 1 is encoded as 10. We will also use Proposition 24, that Fredkin generates CCSWAP.

As in Theorem 23, we can decompose any reversible transformation $F : \{0,1\}^n \to \{0,1\}^n$ as a product of transpositions $\sigma_{y,z}$. In this case, each $\sigma_{y,z}$ transposes two n-bit strings $y = y_1 \ldots y_n$ and $z = z_1 \ldots z_n$ of the same Hamming weight.

Given any n-bit string w, let us define w-CSWAP to be the $(n+2)$-bit gate that swaps its last two bits if its first n bits are equal to w, and that does nothing otherwise. (Thus, Fredkin is 1-CSWAP, while CCSWAP is 11-CSWAP.) Then given y-CSWAP and z-CSWAP gates, where $|y| = |z|$, as well as CCSWAP gates, we can implement the transposition $\sigma_{y,z}$ on input x as follows:

1. Initialize two ancilla bits (comprising one dual-rail register) to $a\overline{a} = 01$.
2. Apply y-CSWAP $(x_1 \ldots x_n, a, \overline{a})$.
3. Apply z-CSWAP $(x_1 \ldots x_n, a, \overline{a})$.
4. Pair off the i's such that $y_i = 1$ and $z_i = 0$, with the equally many j's such that $z_j = 1$ and $y_j = 0$. For each such (i, j) pair, apply Fredkin (a, x_i, x_j).
5. Apply z-CSWAP $(x_1 \ldots x_n, a, \overline{a})$.
6. Apply y-CSWAP $(x_1 \ldots x_n, a, \overline{a})$.

[15] The number of Toffoli gates $T(n)$ needed to implement a 1^n-CNOT (which dominates the cost of a transposition) by this recursive scheme, is given by the recurrence

$$T(n) = 2T(1 + \lfloor n/2 \rfloor) + 2T(\lceil n/2 \rceil)$$

which we solve to obtain $T(n) = O\left(n^2\right)$.

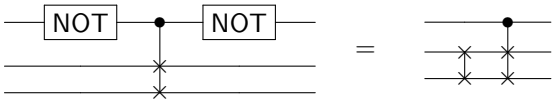


Figure 5 Removing NOT gates from the Fredkin circuit.

The logic here is exactly the same as in the construction of transpositions in Theorem 23; the only difference is that now we need to conserve Hamming weight.

All that remains is to implement w-CSWAP using CCSWAP. First let us show how to implement 1^n-CSWAP using CCSWAP. Once again, we do so using a recursive construction. For the base case, $n = 2$, we just use CCSWAP. For $n \geq 3$, we implement 1^n-CSWAP $(x_1, \ldots, x_n, y, z)$ as follows:

- Initialize two ancilla bits (comprising one dual-rail register) to $a\overline{a} = 01$.
- Apply $1^{\lceil n/2 \rceil}$-CSWAP $\left(x_1 \ldots x_{\lceil n/2 \rceil}, a, \overline{a}\right)$.
- Apply $1^{\lfloor n/2 \rfloor + 1}$-CSWAP $\left(x_{\lceil n/2 \rceil + 1} \ldots x_n, a, y, z\right)$.
- Apply $1^{\lceil n/2 \rceil}$-CSWAP $\left(x_1 \ldots x_{\lceil n/2 \rceil}, a, \overline{a}\right)$.
- Apply $1^{\lfloor n/2 \rfloor + 1}$-CSWAP $\left(x_{\lceil n/2 \rceil + 1} \ldots x_n, a, y, z\right)$.

The logic is the same as in the construction of 1^n-CNOT in Theorem 23 except we now use 2 ancilla bits for the dual-rail representation.

Finally, we need to implement w-CSWAP $(x_1 \ldots x_n, y, z)$, for arbitrary w, using 1^n-CSWAP. We do so by first constructing w-CSWAP from NOT gates and 1^n-CSWAP. Observe that we only use the NOT gate on the control bits of the Fredkin gates used during the construction so the equivalence given in Figure 5 holds (i.e., we can remove the NOT gates).

Hence, we can build a w-CSWAP out of CCSWAPs using only 5 ancilla bits: 1 for CCSWAP, 2 for the 1^n-CSWAP, and 2 for a transposition. ◀

We note that, before the above construction was found by the authors, unpublished and independent work by Siyao Xu and Qian Yu first showed that $O(1)$ ancillas were sufficient and has since been improved to exactly 1 ancilla [31].

In [11], the result that Fredkin generates all conservative transformations is stated without proof, and credited to B. Silver. We do not know how many ancilla bits Silver's construction used.

Next, we prove an analogue of Theorem 23 for the mod-k-respecting transformations, for all $k \geq 2$. First, let CC_k, or Controlled-C_k, be the $(k+1)$-bit gate that applies C_k to the final k bits if the first bit is 1, and does nothing if the first bit is 0.

▶ **Proposition 26.** Fredkin $+ \mathrm{C}_k$ *generates* CC_k*, using* 2 *ancilla bits, for all* $k \geq 2$*.*

Proof. To implement CC_k on input bits $x, y_1 \ldots y_k$, we do the following:

1. Initialize ancilla bits a, b to $0, 1$ respectively.
2. Use Fredkin gates and swaps to swap y_1, y_2 with a, b, conditioned on $x = 0$.[16]
3. Apply C_k to $y_1 \ldots y_k$.
4. Repeat step 2. ◀

Then we have the following.

[16] In more detail, use Fredkin gates to swap y_1, y_2 with a, b, conditioned on $x = 1$. Then swap y_1, y_2 with a, b unconditionally.

▶ **Theorem 27.** Fredkin + CC_k *generates all mod-k-preserving transformations, for* $k \geq 1$*, using only* 5 *ancilla bits.*

Proof. The proof is exactly the same as that of Theorem 25, except for one detail. Namely, let y and z be n-bit strings such that $|y| \equiv |z| \pmod{k}$. Then in the construction of the transposition $\sigma_{y,z}$ from y-CSWAP and z-CSWAP gates, when we are applying step 5, it is possible that $|y| - |z|$ is some nonzero multiple of k, say qk. If so, then we can no longer pair off each i such that $y_i = 1$ and $z_i = 0$ with a unique j such that $z_j = 1$ and $y_j = 0$: after we have done that, there will remain a surplus of '1' bits of size qk, either in y or in z, as well as a matching surplus of '0' bits of size qk in the other string. However, we can get rid of both surpluses using q applications of a CC_k gate (which we have by Proposition 26), with c as the control bit. ◀

As a special case of Theorem 27, note that Fredkin + CC_1 = Fredkin + CNOT generates all mod-1-preserving transformations – or in other words, all transformations.

We just need one additional fact about the C_k gate.

▶ **Proposition 28.** C_k *generates* Fredkin*, using* $k-2$ *ancilla bits, for all* $k \geq 3$*.*

Proof. Let $a_1 \ldots a_{k-2}$ be ancilla bits initially set to 1. Then to implement Fredkin on input bits x, y, z, we apply:

$$\mathrm{C}_k(x, y, a_1 \ldots a_{k-2}), \mathrm{C}_k(x, z, a_1 \ldots a_{k-2}), \mathrm{C}_k(x, y, a_1 \ldots a_{k-2}).$$ ◀

Combining Theorem 27 with Proposition 28 now yields the following.

▶ **Corollary 29.** C_k *generates all mod-k-preserving transformations for* $k \geq 3$*, using only* $k+3$ *ancilla bits.*

Finally, we handle the parity-flipping case.

▶ **Proposition 30.** Fredkin + NOTNOT *(and hence,* Fredkin + NOT*) generates* CC_2*.*

Proof. This follows from Proposition 26, if we recall that C_2 is equivalent to NOTNOT up to an irrelevant bit-swap. ◀

▶ **Theorem 31.** Fredkin + NOT *generates all parity-respecting transformations, using only* 5 *ancilla bits.*

Proof. Let F be any parity-flipping transformation, and let F' be F followed by NOT on one of the output bits. Then F' is parity-preserving. So by Theorem 27, we can implement F' using Fredkin + CC_2 (and we have CC_2 by Proposition 30). We can then apply another NOT gate to get F itself. ◀

One consequence of Theorem 31 is that every parity-flipping transformation can be constructed from parity-preserving gates and exactly one NOT gate.

7.2 Affine Circuits

It is well-known that CNOT is a "universal affine gate":

▶ **Theorem 32.** CNOT *generates all affine transformations, with only* 1 *ancilla bit (or* 0 *for linear transformations).*

Proof. Let $G(x) = Ax \oplus b$ be the affine transformation that we want to implement, for some invertible matrix $A \in \mathbb{F}_2^{n \times n}$. Then given an input $x = x_1 \dots x_n$, we first use CNOT gates (at most $\binom{n}{2}$ of them) to map x to Ax, by reversing the sequence of row-operations that maps A to the identity matrix in Gaussian elimination. Finally, if $b = b_1 \dots b_n$ is nonzero, then for each i such that $b_i = 1$, we apply a CNOT from an ancilla bit that is initialized to 1. ◀

A simple modification of Theorem 32 handles the parity-preserving case.

▶ **Theorem 33.** CNOTNOT *generates all parity-preserving affine transformations with only* 1 *ancilla bit (or* 0 *for linear transformations).*

Proof. Let $G(x) = Ax \oplus b$ be a parity-preserving affine transformation. We first construct the linear part of G using Gaussian elimination. Notice that for G to be parity-preserving, the columns v_i of A must satisfy $|v_i| \equiv 1 \pmod 2$ for all i. For this reason, the row-elimination steps come in pairs, so we can implement them using CNOTNOT. Notice further that since G is parity-preserving, we must have $|b| \equiv 0 \pmod 2$. So we can map Ax to $Ax \oplus b$, by using CNOTNOT gates plus one ancilla bit set to 1 to simulate NOTNOT gates. ◀

Likewise (though, strictly speaking, we will not need this for the proof of Theorem 3):

▶ **Theorem 34.** CNOTNOT + NOT *generates all parity-respecting affine transformations using no ancilla bits.*

Proof. Use Theorem 33 to map x to Ax, and then use NOT gates to map Ax to $Ax \oplus b$. ◀

We now move on to the more complicated cases of $\langle \mathrm{F}_4 \rangle$, $\langle \mathrm{T}_6 \rangle$, and $\langle \mathrm{T}_4 \rangle$.

▶ **Theorem 35.** F_4 *generates all mod-4-preserving affine transformations using no ancilla bits.*

Proof. Let $F(x) = Ax \oplus b$ be an n-bit affine transformation, $n \geq 2$, that preserves Hamming weight mod 4. Using F_4 gates, we will show how to map $F(x) = y_1 \dots y_n$ to $x = x_1 \dots x_n$. Reversing the construction then yields the desired map from x to $F(x)$.

At any point in time, each y_j is some affine function of the x_i's. We say that x_i "occurs in" y_j, if y_j depends on x_i. At a high level, our procedure will consist of the following steps, repeated up to $n - 3$ times:

1. Find an x_i that does not occur in every y_j.
2. Manipulate the y_j's so that x_i occurs in exactly *one* y_j.
3. Argue that no *other* $x_{i'}$ can then occur in that y_j. Therefore, we have recursively reduced our problem to one involving a reversible, mod-4-preserving, affine function on $n - 1$ variables.

It is not hard to see that the only mod-4-preserving affine functions on 3 or fewer variables, are permutations of the bits. So if we can show that the three steps above can always be carried out, then we are done.

First, since A is invertible, it is not the all-1's matrix, which means that there must be an x_i that does not occur in every y_j.

Second, if there are at least three occurrences of x_i, then apply F_4 to three positions in which x_i occurs, plus one position in which x_i does not occur. The result of this is to decrease the number of occurrences of x_i by 2. Repeat until there are at most two occurrences of x_i. Since F_4 is mod-4-preserving and affine, the resulting transformation $F'(x) = A'x + b'$ must still be mod-4-preserving and affine, so it must still satisfy the conditions of Lemma 20. In

particular, no column vector of A' can have even Hamming weight. Since two occurrences of x_i would necessitate such a column vector, we know that x_i must occur only once.

Third, if x_i occurs only once in $F'(x)$, then the corresponding column vector v_i has exactly one nonzero element. Since $|v_i| = 1$, we know by Lemma 20 that $v_i \cdot b \equiv 0 \pmod 2$, which means that b has a 0 in the position where v_i has a 1. Now consider the row of A' that includes the nonzero entry of v_i. If any other column $v_{i'}$ is also nonzero in that row, then $v_i \cdot v_{i'} \equiv 1 \pmod 2$, which once again contradicts the conditions of Lemma 20. Thus, no other $x_{i'}$ occurs in the same y_j that x_i occurs in. Indeed no constant occurs there either, since otherwise F' would no longer be mod-4-preserving. So we have reduced to the $(n-1) \times (n-1)$ case. ◀

The same argument, with slight modifications, handles $\langle \mathrm{T}_4 \rangle$ and $\langle \mathrm{T}_6 \rangle$.

▶ **Theorem 36.** T_4 *generates all orthogonal transformations, using no ancilla bits.*

Proof. The construction is identical to that of Theorem 35, except with T_4 instead of F_4. When reducing the number of occurrences of x_i to at most 2, Lemma 16 assures us that $|v_i| \equiv 1 \pmod 2$. ◀

▶ **Theorem 37.** T_6 *generates all mod-4-preserving linear transformations, using no ancilla bits.*

Proof. The construction is identical to that of Theorem 35, except for the following change. Rather than using F_4 to reduce the number of occurrences of some x_i to at most 2, we now use T_6 to reduce the number of occurrences of x_i to at most 4. (If there are 5 or more occurrences, then T_6 can always decrease the number by 4.) We then appeal to Corollary 22, which says that $|v_i| \equiv 1 \pmod 4$ for each i. This implies that no x_i can occur 2, 3, or 4 times in the output vector. But that can only mean that x_i occurs once. ◀

By Lemma 14 and Corollary 21, an equivalent way to state Theorem 37 is that T_6 generates all affine transformations that are both mod-4-preserving and orthogonal.

All that remains is some "cleanup work" (which, again, is not even needed for the proof of Theorem 3).

▶ **Theorem 38.** $\mathrm{T}_6 + \mathrm{NOT}$ *generates all affine transformations that are mod-4-preserving (and therefore orthogonal) in their linear part.*

$\mathrm{T}_6 + \mathrm{NOTNOT}$ *generates all parity-preserving affine transformations that are mod-4-preserving (and therefore orthogonal) in their linear part.*

$\mathrm{F}_4 + \mathrm{NOT}$ *(or equivalently,* $\mathrm{T}_4 + \mathrm{NOT}$*) generates all isometries.*

$\mathrm{F}_4 + \mathrm{NOTNOT}$ *(or equivalently,* $\mathrm{T}_4 + \mathrm{NOTNOT}$*) generates all parity-preserving isometries.*

NOT *generates all degenerate transformations.*

NOTNOT *generates all parity-preserving degenerate transformations.*

In none of these cases are any ancilla bits needed.

Proof. As in Theorem 34, we simply apply the relevant construction for the linear part (e.g., Theorem 36 or 37), then handle the affine part using NOT or NOTNOT gates. ◀

8 Open Problems

As discussed in Section 1, the central challenge we leave is to give a complete classification of all *quantum* gate sets acting on qubits, in terms of which unitary transformations they can generate or approximate. Here, just like in this paper, one should assume that qubit-swaps are free, and that arbitrary ancillas are allowed as long as they are returned to their initial states.

A possible first step in the direction we want, which would involve Lie algebras, would be to classify all sets of 1- and 2-qubit gates. A second step would be to classify qubit Hamiltonians (i.e., the infinitesimal-time versions of unitary gates), in terms of which n-qubit Hamiltonians they can be used to generate. Here the recent work of Cubitt and Montanaro [10], which classifies qubit Hamiltonians in terms of the complexity of approximating ground state energies, might be relevant. Yet a third possibility would be to classify quantum gates under the assumption that intermediate measurements are allowed. Of course, these simplifications can also be combined.

On the classical side, we have left completely open the problem of classifying reversible gate sets over non-binary alphabets. In the non-reversible setting, it was discovered in the 1950s (see [20]) that Post's lattice becomes dramatically different and more complicated when we consider gates over a 3-element set rather than Boolean gates: for example, there is now an uncountable infinity of clones, rather than "merely" a countable infinity. Does anything similar happen in the reversible case? We know from Gu [14] that for alphabets of size greater than three, there exists a class that is not finitely generated. Recall that for binary alphabets, one gate always suffices to generate a particular class.

Even for reversible gates over (say) $\{0,1,2\}^n$, we cannot currently give an algorithm to decide whether a given gate G generates another gate H any better than the triple-exponential-time algorithm that comes from clone theory, nor can we give reasonable upper bounds on the number of gates or ancillas needed in the generating circuit, nor can we answer basic questions like whether every class is finitely generated.

Finally, can one reduce the number of gates in each of our circuit constructions to the limits imposed by Shannon-style counting arguments? What are the tradeoffs, if any, between the number of gates and the number of ancilla bits?

Acknowledgments. At the very beginning of this project, Emil Jeřábek [15] brought the $\langle \mathrm{C}_k \rangle$ and $\langle \mathrm{T}_6 \rangle$ classes to our attention, and also proved that every reversible gate class is characterized by invariants (i.e., that the "clone-coclone duality" holds for reversible gates). Also, Matthew Cook gave us encouragement, asked pertinent questions, and helped us understand the $\langle \mathrm{T}_4 \rangle$ class. We are grateful to both of them. We also thank Harry Altman, Adam Bouland, Seth Lloyd, Igor Markov, and particularly Siyao Xu for helpful discussions.

References

1 S. Aaronson and A. Arkhipov. The computational complexity of linear optics. *Theory of Computing*, 9(4):143–252, 2013. Conference version in Proceedings of ACM STOC 2011. ECCC TR10-170, arXiv:1011.3245.

2 S. Aaronson and A. Bouland. Generation of universal linear optics by any beam splitter. *Phys. Rev. A*, 89(6):062316, 2014. arXiv:1310.6718.

3 S. Aaronson and D. Gottesman. Improved simulation of stabilizer circuits. *Phys. Rev. A*, 70(052328), 2004. arXiv:quant-ph/0406196.

4 Scott Aaronson, Daniel Grier, and Luke Schaeffer. The classification of reversible bit operations. *arXiv:1504.05155*, 2015.

5 D. Bacon, J. Kempe, D. P. DiVincenzo, D. A. Lidar, and K. B. Whaley. Encoded universality in physical implementations of a quantum computer. In R. Clark, editor, *Proceedings of the 1st International Conference on Experimental Implementations of Quantum Computation*, page 257. Rinton, 2001. arXiv:quant-ph/0102140.

6 A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, 52(3457), 1995. arXiv:quant-ph/9503016.

7 M. Ben-Or and R. Cleve. Computing algebraic formulas with a constant number of registers. In *Proc. ACM STOC*, pages 254–257, 1988.

8 C. H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17:525–532, 1973.

9 R. Cleve and J. Watrous. Fast parallel circuits for the quantum Fourier transform. In *Proc. IEEE FOCS*, pages 526–536, 2000. arXiv:quant-ph/0006004.

10 T. Cubitt and A. Montanaro. Complexity classification of local Hamiltonian problems. In *Proc. IEEE FOCS*, pages 120–129, 2014. arXiv:1311.3161.

11 E. Fredkin and T. Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21(3-4):219–253, 1982.

12 D. Gottesman. Class of quantum error-correcting codes saturating the quantum Hamming bound. *Phys. Rev. A*, 54:1862–1868, 1996. arXiv:quant-ph/9604038.

13 D. Grier and L. Schaeffer. The classification of stabilizer operations over qubits. *ArXiv e-prints*, March 2016. arXiv:1603.03999.

14 Y. Gu. Some results on reversible gate classes over non-binary alphabets. *CoRR*, abs/1606.00804, 2016. URL: `http://arxiv.org/abs/1606.00804`.

15 E. Jeřábek. Answer to CS Theory StackExchange question on "classifying reversible gates". At `http://cstheory.stackexchange.com/questions/25730/classifying-reversible-gates`, 2014.

16 P. Kerntopf, M. A. Perkowski, and M. Khan. On universality of general reversible multiple-valued logic gates. In *IEEE International Symposium on Multiple-Valued Logic*, pages 68–73, 2004.

17 O. G. Kharlampovich and M. V. Sapir. Algorithmic problems in varieties. *International Journal of Algebra and Computation*, 5(04n05):379–602, 1995. `http://www.math.vanderbilt.edu/~msapir/ftp/pub/survey/survey.pdf`.

18 E. Knill and R. Laflamme. Power of one bit of quantum information. *Phys. Rev. Lett.*, 81(25):5672–5675, 1998. arXiv:quant-ph/9802037.

19 R. Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3):183–191, 1961.

20 D. Lau. *Function Algebras on Finite Sets: Basic Course on Many-Valued Logic and Clone Theory*. Springer, 2006.

21 S. Lloyd. Any nonlinear one-to-one binary logic gate suffices for computation. Technical Report LA-UR-92-996, Los Alamos National Laboratory, 1992. arXiv:1504.03376.

22 K. Morita, T. Ogiro, K. Tanaka, and H. Kato. Classification and universality of reversible logic elements with one-bit memory. In *Proceedings of the 4th International Conference on Machines, Computations, and Universality*, pages 245–256. Springer-Verlag, 2005.

23 E. L. Post. *The two-valued iterative systems of mathematical logic*. Number 5 in Annals of Mathematics Studies. Princeton University Press, 1941.

24 M. Saeedi and I. L. Markov. Synthesis and optimization of reversible circuits–a survey. *ACM Computing Surveys*, 45(2):21, 2013. arXiv:1110.2574.

25 L. Schaeffer. Reversible Gate Classifier, 2015. https://github.com/lrschaeffer/Gate-Classifier.

26 V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes. Synthesis of reversible logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(6):710–722, 2003. arXiv:quant-ph/0207001.

27 Y. Shi. Both Toffoli and controlled-NOT need little help to do universal quantum computation. *Quantum Information and Computation*, 3(1):84–92, 2002. quant-ph/0205115.

28 I. Strazdins. Universal affine classification of Boolean functions. *Acta Applicandae Mathematica*, 46(2):147–167, 1997.

29 T. Toffoli. Reversible computing. In *Proc. Intl. Colloquium on Automata, Languages, and Programming (ICALP)*, pages 632–644. Springer, 1980.

30 A. De Vos and L. Storme. r-universal reversible logic gates. *Journal of Physics A: Mathematical and General*, 37(22):5815–5824, 2004.

31 S. Xu. Reversible logic synthesis with minimal usage of ancilla bits. *CoRR*, abs/1506.03777, 2015. URL: http://arxiv.org/abs/1506.03777.

Nondeterministic Quantum Communication Complexity: the Cyclic Equality Game and Iterated Matrix Multiplication*

Harry Buhrman[1], Matthias Christandl[2], and Jeroen Zuiddam[3]

1 **QuSoft, CWI and University of Amsterdam, Amsterdam, The Netherlands**
buhrman@cwi.nl
2 **QMATH, Department of Mathematical Sciences, University of Copenhagen, Copenhagen, Denmark**
christandl@math.ku.dk
3 **QuSoft, CWI and University of Amsterdam, Amsterdam, The Netherlands**
j.zuiddam@cwi.nl

Abstract

We study nondeterministic multiparty quantum communication with a quantum generalization of broadcasts. We show that, with number-in-hand classical inputs, the communication complexity of a Boolean function in this communication model equals the logarithm of the *support rank* of the corresponding tensor, whereas the approximation complexity in this model equals the logarithm of the *border support rank*. This characterisation allows us to prove a log-rank conjecture posed by Villagra et al. for nondeterministic multiparty quantum communication with message passing.

The support rank characterization of the communication model connects quantum communication complexity intimately to the theory of asymptotic entanglement transformation and algebraic complexity theory. In this context, we introduce the *graphwise equality problem*. For a cycle graph, the complexity of this communication problem is closely related to the complexity of the computational problem of multiplying matrices, or more precisely, it equals the logarithm of the support rank of the iterated matrix multiplication tensor. We employ Strassen's laser method to show that asymptotically there exist nontrivial protocols for every odd-player cyclic equality problem. We exhibit an efficient protocol for the 5-player problem for small inputs, and we show how Young flattenings yield nontrivial complexity lower bounds.

1998 ACM Subject Classification E.4 Coding and Information Theory

Keywords and phrases Quantum communication complexity, broadcast channel, number-in-hand, matrix multiplication, support rank

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.24

1 Introduction

Let $f : X \times Y \times Z \to \{0, 1\}$ be a function on a product of finite sets X, Y and Z. Alice, Bob and Charlie have to compute f in the following sense. Alice receives an $x \in X$, Bob

* Part of this work was done while MC and JZ were visiting the Simons Institute for the Theory of Computing, UC Berkeley. HB was partially funded by the European Commission, through the SIQS project and by the Netherlands Organisation for Scientific Research (NWO) through gravitation grant Networks. MC acknowledges financial support from the European Research Council (ERC Grant Agreement no 337603), the Danish Council for Independent Research (Sapere Aude) and VILLUM FONDEN via the QMATH Centre of Excellence (Grant No. 10059). Part of this work was done while MC was with ETH Zurich. JZ is supported by NWO through the research programme 617.023.116 and by the European Commission through the SIQS project


© Harry Buhrman, Matthias Christandl, and Jeroen Zuiddam;
licensed under Creative Commons License CC-BY

receives a $y \in Y$ and Charlie receives a $z \in Z$, and each player receives a private random bit string. Then the players communicate in rounds. Each round, one player communicates by broadcasting a bit to the other players. After these rounds of communication, each player has to output a bit, such that if $f(x, y, z) = 1$, then with some nonzero probability all players output 1 and if $f(x, y, z) = 0$, then with probability zero all players output 1. The complexity of such a protocol is the number of broadcasts in the protocol, and we denote the minimum complexity of all such protocols by $\mathrm{N}(f)$.

Now we allow the players to be quantum, as follows. Alice receives an $x \in X$, Bob receives a $y \in Y$ and Charlie receives a $z \in Z$. Then, the players communicate by creating a GHZ state of rank r

$$|\mathrm{GHZ}_r\rangle = \tfrac{1}{\sqrt{r}}(|111\rangle + |222\rangle + \cdots + |rrr\rangle).$$

and sharing this state among each other, a quantum broadcast. Next, the players do local quantum operations. Finally, each player has to output a bit, such that if $f(x, y, z) = 1$, then with some nonzero probability all players output 1 and if $f(x, y, z) = 0$, then with probability zero all players output 1. The quantum complexity of such a quantum protocol is $\log_2 r$, and we denote the minimum complexity of all quantum protocols by $\mathrm{NQ}(f)$. We will make this definition more precise and more general in Section 2. Note that the quantum model can simulate the classical model by a postselection procedure. Also note that, nondeterministically, one quantum broadcast can be used to send a qubit from one player to another by using teleportation (see Theorem 8); the quantum model can thus simulate a message-passing model. The classical and quantum communication model naturally extend to k players.

1.1 Our results

- Our main technical result is that the quantum complexity of a function in the above model equals the logarithm of the so-called *support rank* of the tensor $\sum_{x,y,z} f(x, y, z)\,|x\rangle|y\rangle|z\rangle$ corresponding to f. We prove this in Section 2.
- Modifying the quantum model such that the players can only communicate by message passing and there is no shared $|\mathrm{GHZ}_r\rangle$ at the start – that is, the players now communicate in rounds and in each communication round one player sends a qubit to one other player – increases the complexity by at most a factor $k - 1$ (with k the number of players), and this relationship is tight. However, asymptotically in the input size, the increase is only $k/2$ and this relationship is tight. This solves a *nondeterministic multiplayer quantum log-rank conjecture* in the message-passing model of Villagra et al. [19]. This topic is covered in Section 3.
- We define the k-player *graphwise equality* problem to be the problem in which k players are identified with vertices in a graph G, and each player has to compute the equality function with his neighbours in G. Of particular interest is the cycle graph $G = C_k$ and the corresponding *cyclic equality* problem. For this cyclic equality problem, in the classical broadcast model, the naïve protocol in which every player broadcasts his inputs is the optimal protocol. The same holds in the quantum model when k is even. Interestingly, we show with Strassen's laser method that for all odd $k \geq 3$ there is a nontrivial quantum protocol. Moreover, for all odd $k \geq 3$ we give nontrivial lower bounds on the value of NQ by use of Young flattenings. These results are related to the complexity of matrix multiplication and iterated matrix multiplication. In particular, we improve a lower bound of Ikenmeyer on the border support rank of IMM_n^3 [11, 8.2.17]. A consequence of our work is that finding new protocols for the cyclic equality problem for three players yields new algorithms for matrix multiplication. Section 4 covers the classical case, the even quantum case, an explicit quantum protocol for $k = 5$, and the Young flattening lower bound. Section 5 covers the Strassen laser method.

1.2 Related work

The two-player nondeterministic quantum communication model was introduced by De Wolf [21]. He shows that the communication complexity in this model is characterized by the logarithm of the support rank of the communication matrix. The quantum broadcast channel, a communication model that is very similar to ours, has been studied by e.g. Ambainis et al. [1]. Multiparty nondeterministic quantum communication with message passing has been studied by Villagra et al. [19]. They show that the logarithm of the support rank of the communication tensor is a lower bound for the message-passing complexity and conjecture that this lower bound is polynomially related to the message-passing complexity.

The support rank of 3-tensors has been studied by Cohn and Umans in the context of the complexity of matrix multiplication [9]. They give nontrivial upper bounds on the support rank of the matrix multiplication tensor that do not come from upper bounds on the tensor rank. As an interesting fact, we note that given a matrix A and a number k, deciding whether the support rank of A is at least k is NP-hard [3].

The complexity of matrix multiplication plays a central role in algebraic complexity theory. We refer to [6] for general background information. Connections between algebraic complexity theory and entanglement transformations have been studied before, see for example [7]. The iterated matrix multiplication tensor has been studied in the context of arithmetic circuit complexity and the VP versus VNP problem, see for example [10]. To the knowledge of the authors, the tensor rank or support rank of the iterated matrix multiplication tensor has not been studied before.

Our work has motivated the further investigation of the tensor rank tensors defined by cycle graphs and more general graphs [8], which in turn when used in conjunction with this paper, lead to improved bounds on the non-deterministic quantum communication complexity of the cyclic equality game and more generally equality games played on graphs.

2 Support rank characterization of the quantum broadcast model

We refer to Nielsen and Chuang [15] for background information on the quantum computation model.

Quantum multiparty communication protocol. For any natural number m, denote by $[m]$ the set $\{1, 2, \ldots, m\}$. Let k be a positive integer and let f be a Boolean function on $[2^n]^k = [2^n] \times [2^n] \times \cdots \times [2^n]$,

$$f : [2^n]^k \to \{0, 1\}.$$

We define a *k-player quantum communication protocol* as follows. Each player i has a finite-dimensional Hilbert space H_i. The protocol thus takes place in the space $H_1 \otimes \cdots \otimes H_k$. The space is initialised in the state $|x_1 \cdots x_k\rangle |\mathrm{GHZ}_r^k\rangle$, where

$$|\mathrm{GHZ}_r^k\rangle \coloneqq \sum_{a=1}^{r} |a\rangle|a\rangle \cdots |a\rangle \in (\mathbf{C}^r)^{\otimes k}$$

is the k-party GHZ-state of rank r, shared among the k players, and $x_i \in [2^n]$ is the classical input to player i. (For clarity we will suppress any normalizations in quantum states when possible.) The players now apply local quantum operations. Let R_i be the first qubit of H_i and let $R = R_1 \otimes \cdots \otimes R_k$. We apply a projection onto $|11 \cdots 1\rangle$ in R. If the resulting tensor is 0 then the output of the protocol is 0, otherwise the output of the protocol is 1. The

complexity of the protocol is $\log_2(r)$. We say the protocol *nondeterministically computes* f if the probability that the output equals 1 is nonzero if $f(x_1, \ldots, x_k) = 1$ and the probability that the output equals 0 is one if $f(x_1, \ldots, x_k) = 0$.

▶ **Definition 1.** Let k be a positive integer and let f be a function $[2^n]^k \to \{0, 1\}$. The *k-player nondeterministic quantum communication complexity of f* is the minimal complexity of a k-player quantum communication protocol that nondeterministically computes f, and is denoted by $\mathrm{NQ}(f)$.

Approximating protocols. Let f be a function $[2^n]^k \to \{0, 1\}$. Let $(\Pi_j)_{j \in \mathbf{N}}$ be a sequence of protocols, such that when $f(x_1, \ldots, x_k) = 1$, the probability that Π_j outputs 1 on input x converges to a nonzero number as j goes to infinity, and when $f(x_1, \ldots, x_k) = 0$, the probability that Π_j outputs 0 on input x converges to 1 as j goes to infinity. Then we say that the sequence $(\Pi_j)_{j \in \mathbf{N}}$ *approximately nondeterministically computes* f. The complexity of an approximating sequence is the maximum complexity of any protocol Π_j in the sequence.

▶ **Definition 2.** The *k-player approximate nondeterministic quantum communication complexity of f* is the minimal complexity of a sequence (Π_j) that approximately nondeterministically computes f, and is denoted by $\underline{\mathrm{NQ}}(f)$.

Classical protocol. We define a *k-player classical communication protocol* as follows. Each player receives a classical input and a private random bit string. The protocol proceeds in rounds. Each round we let a single predetermined player communicate by broadcasting a bit to all the other players. After the last communication round, every player presents an output bit. If all the output bits are 1, then the output of the protocol is 1; otherwise the output of the protocol is 0. Again, we say the classical protocol *nondeterministically computes* f if the probability that the output equals 1 is nonzero if $f(x_1, \ldots, x_k) = 1$ and the probability that the output equals 0 is one if $f(x_1, \ldots, x_k) = 0$.

▶ **Definition 3.** The *k-player nondeterministic classical communication complexity of f* is the minimal complexity of a k-player classical communication protocol that nondeterministically computes f, and is denoted by $\mathrm{N}(f)$.

▶ Remark. For simplicity, we have taken the input set for each of the k players to be the same set $[2^n]$. We note that the definitions in this section and most of the results in this paper naturally generalize to the situation where the players get inputs from sets of different sizes.

Support rank and border support rank. Let t be a tensor in $(\mathbf{C}^m)^{\otimes k}$. The *tensor rank* of t is the smallest number r such that t can be written as a sum of r simple tensors, that is, $t = \sum_{i=1}^{r} u_i^1 \otimes u_i^2 \otimes \cdots \otimes u_i^k$ for some vectors $u_i^j \in \mathbf{C}^m$. We denote the tensor rank of t by $\mathrm{R}(t)$. Let $|1\rangle, \ldots, |m\rangle$ be the standard basis for $\mathbf{C}^m$ and define the support of a tensor t in $(\mathbf{C}^m)^{\otimes k}$ to be the set of product basis elements $|i_1\rangle \otimes \cdots \otimes |i_k\rangle$ that occur with nonzero coefficient in t. The *support rank* or *nondeterministic rank* of t is the smallest number r such that there exists a tensor in the space $(\mathbf{C}^m)^{\otimes k}$ with the same support as t and tensor rank r. We denote the support rank of t by $\mathrm{R_s}(t)$. Note that support rank is basis *dependent*.

The *border rank* of t is the smallest number r such that there exists a sequence of tensors $(t_j)_{j \in \mathbf{N}}$ converging to t in the Euclidean topology (or equivalently in the Zariski topology) such that $\mathrm{R}(t_j)$ is at most r for every j. We denote the border rank of t by $\underline{\mathrm{R}}(t)$. The *border support rank* of t is the smallest number r such that there exists a tensor in $(\mathbf{C}^m)^{\otimes k}$ with the same support as t and border rank r. We denote the border support rank of t by $\underline{\mathrm{R}}_{\mathrm{s}}(t)$.

▶ **Theorem 4.** *Let $f : [2^n]^k \to \{0,1\}$ be a function and let t be the tensor in $(\mathbf{C}^{2^n})^{\otimes k}$ with entries given by f, that is, $t = \sum_{i \in [2^n]^k} f(i)\,|i_1\rangle|i_2\rangle \cdots |i_k\rangle$. Then $\mathrm{NQ}(f) = \log_2 \mathrm{R_s}(t)$ and $\underline{\mathrm{NQ}}(f) = \log_2 \underline{\mathrm{R}}_{\mathrm{s}}(t)$.*

▶ **Lemma 5** (Cleanup lemma)**.** *Let $\{|\psi_i\rangle : i \in [q]\} \subseteq (\mathbf{C}^m)^{\otimes k}$ be a set of k-tensors, for some natural number q. Then there exists a k-partite rank-1 linear map $\langle \ell| \coloneqq \langle \ell_1| \otimes \cdots \otimes \langle \ell_k|$ with $\langle \ell_j| \in (\mathbf{C}^m)^*$ such that $\langle \ell|\psi_i\rangle \neq 0$ for every $i \in [q]$.*

Proof. We will give a proof by recursively constructing $\langle \ell|$. Let Id be the identity map on $\mathbf{C}^m$. If $j \leq k$, $\langle a| \in ((\mathbf{C}^m)^*)^{\otimes j}$ and $|b\rangle \in (\mathbf{C}^m)^{\otimes k}$, then we denote by $\langle a|b\rangle$ the contraction of $\langle a|$ and $|b\rangle$, that is, $\langle a|b\rangle = (\langle a| \otimes \mathrm{Id}^{\otimes k-j})\,|b\rangle$.

The base case is $\langle \ell| = 1$. For the recursion, suppose we are given an element $\langle \ell'| \in ((\mathbf{C}^m)^*)^{\otimes j}$ such that $|\phi_i\rangle \coloneqq \langle \ell'|\psi_i\rangle$ is nonzero for every $i \in [q]$. We will construct an element $\langle \ell| \in ((\mathbf{C}^m)^*)^{\otimes j+1}$ such that $\langle \ell|\psi_i\rangle$ is nonzero for every $i \in [q]$. Since $|\phi_i\rangle$ is nonzero for every $i \in [q]$, there is an element $\langle u_i| \in (\mathbf{C}^m)^*$ such that $\langle u_i|\phi_i\rangle$ is nonzero. Consider the the maps $(\langle u_1| + x\,\langle u_2|)\,|\phi_i\rangle$ for $i \in \{1,2\}$, in the variable x. Each map only has a single root. Therefore, there exists a value α_2 for x such that both maps evaluate to a nonzero number. Next, consider the maps $(\langle u_1| + \alpha_2\,\langle u_2| + x\,\langle u_3|)\,|\phi_i\rangle$ for $i \in \{1,2,3\}$, in variable x. Again, each of the three maps has only a single root. Therefore, there exists a value α_3 for x such that all three maps evaluate to a nonzero number. Repeat this construction to obtain an element $\langle u| \in (\mathbf{C}^m)^*$ such that $\langle u|\phi_i\rangle$ is nonzero for every $i \in [q]$. Let $\langle \ell|$ be $\langle \ell'| \otimes \langle u|$. ◀

Proof of Theorem 4. We first show $\mathrm{NQ}(f) \leq \log_2 \mathrm{R_s}(t)$. Let r be the support rank of t. Then there exists a unit vector $\psi \in (\mathbf{C}^{2^n})^{\otimes k}$ with rank r and support equal to the support of f. This means that there are vectors $|u_i^j\rangle \in \mathbf{C}^{2^n}$ such that $\psi = \sum_{i=1}^r |u_i^1\rangle \cdots |u_i^k\rangle$. For every player j define a matrix

$$A_j \coloneqq \alpha_j \sum_{i=1}^{r} |u_i^j\rangle\langle i|$$

where α_j is a nonzero complex number such that $A_j^\dagger A_j$ has eigenvalue at most 1. The matrix $I - A_j^\dagger A_j$ is thus positive semidefinite and hence there exists a matrix A_j' such that ${A_j'}^\dagger A_j' = I - A_j^\dagger A_j$. Define for every player j a quantum operation

$$\mathcal{E}_j : \rho \mapsto A_j \rho A_j^\dagger \otimes |1\rangle\langle 1| + A_j' \rho {A_j'}^\dagger \otimes |0\rangle\langle 0|\,.$$

Note that this operation introduces a new control qubit register which player j can measure to see whether he applied A_j or A_j'.

The protocol for the k players is as follows. Let $x_1, \ldots, x_k$ be the inputs given to the players. The players share a k-party GHZ-state of rank r. Player j applies $\mathcal{E}_j$ to his part of the GHZ-state. If his control qubit is $|0\rangle$ then he sets his output qubit R_i to $|0\rangle$. Otherwise, he measures the rest of the system. If the outcome equals $|x_j\rangle$, then he sets R_j to $|1\rangle$, otherwise he sets R_j to $|0\rangle$.

The above protocol uses a GHZ-state of rank r, so it has complexity $\log_2(r)$. We claim that the protocol nondeterministically computes f. If the players in the first measurement each get outcome $|1\rangle$, then the state of the total system is $|\psi\rangle$. Because $|\psi\rangle$ has norm 1, this happens with nonzero probability $|\alpha_1|^2 \cdots |\alpha_k|^2$. If $f(x_1, \ldots, x_k) = 0$, then $|x_1 \cdots x_k\rangle$ does not occur in the support of ψ, so the probability that the players measure $|x_1\rangle, \ldots, |x_k\rangle$ respectively is zero. Hence in this case the register R is not in state $|11\cdots1\rangle$. On the other hand, if $f(x_1, \ldots, x_k) \neq 0$, then $|x_1 \cdots x_k\rangle$ does occur in the support of ψ, so the probability that the players measure $|x_1\rangle, \ldots, |x_k\rangle$ respectively is nonzero. Hence with nonzero probability the register R is in state $|11\cdots1\rangle$.

We now show $\mathrm{NQ}(f) \geq \log_2 \mathrm{R_s}(t)$. Suppose we have a protocol that nondeterministically computes f with complexity r. This means that the players perform *local* quantum operations that together form a linear map L which transforms, for any $x_1, \ldots, x_k \in [2^n]$, the state

$$|x_1 \cdots x_k\rangle \, |\mathrm{GHZ}_r\rangle$$

to a state of the form

$$|x_1 \cdots x_k\rangle \sum_{a \in \{0,1\}^k} |\psi_x^a\rangle \; |a_1\rangle |a_2\rangle \cdots |a_k\rangle \,,$$

where $|\psi_x^a\rangle$ is some vector representing the state of the work space of the players. By definition of nondeterministic computation, for $a = (1, \ldots, 1)$, if $f(x_1, \ldots, x_k) = 1$, then $|\psi_x^a\rangle$ is nonzero, and if $f(x_1, \ldots, x_k) = 0$, then $|\psi_x^a\rangle$ is zero. Since the map L is linear, it maps the tensor

$$s_1 \coloneqq \sum_{x_1, \ldots, x_k} |x_1 \cdots x_k\rangle \, |\mathrm{GHZ}_r\rangle$$

to the tensor

$$s_2 \coloneqq \sum_{x_1, \ldots, x_k} |x_1 \cdots x_k\rangle \sum_{a \in \{0,1\}^k} |\psi_x^a\rangle \, |a_1 \cdots a_k\rangle \,.$$

The tensor rank of $\sum_x |x_1 \cdots x_k\rangle$ is 1 and hence the tensor rank of s_1 is r. Because L is a local map, the tensor rank of s_2 is at most r. By applying the cleanup lemma (Lemma 5) and projecting on states with $|a_1 \cdots a_k\rangle = |1 \cdots 1\rangle$, we obtain a tensor

$$s_3 \coloneqq \sum_{x_1, \ldots, x_k} |x_1 \cdots x_k\rangle \, c_x$$

where $c_x \in \mathbf{C}$ is zero if $f(x) = 0$ and nonzero if $f(x) = 1$. The rank of the tensor s_3 is at most r. The support of s_3 equals the support of f, so the support rank of f is at most r.

The statement about the approximate complexity of f follows from the definition of border support rank. ◀

▶ **Definition 6** (SLOCC)**.** Let $\phi \in U_1 \otimes \cdots \otimes U_k$ and let $\psi \in V_1 \otimes \cdots \otimes V_k$. We say that ϕ can be converted to ψ by stochastic local operations and classical communication (SLOCC), and write $\phi \xrightarrow{\mathrm{SLOCC}} \psi$, if there exist matrices $A_1, \ldots, A_k$ such that $\psi = (A_1 \otimes \cdots \otimes A_k)\phi$.

▶ Remark. We note that having an NQ-protocol for f of complexity n is the same as having an SLOCC protocol for transforming $\mathrm{GHZ}_{2^n}^k$ to a tensor with the same support as f. We will use the SLOCC paradigm in some parts of the text.

▶ Remark. The NQ-model that we are using is very similar to the following *broadcast channel* model that was studied in [1]. Each player i has a local Hilbert space H_i with a register initialised in the input state $|x_i\rangle$. The players have access to a quantum broadcast channel, which, given a qubit state $\alpha |0\rangle + \beta |1\rangle$, will create the state $\alpha |0\rangle^{\otimes k} + \beta |1\rangle^{\otimes k}$ and distribute this state among the k players. The players proceed in communication rounds; each round a designated player uses the broadcast channel. Let R_i be the first qubit of H_i and let $R = R_1 \otimes \cdots \otimes R_k$. After the communication is finished, we apply a projection onto $|11 \cdots 1\rangle$ in R. If the resulting tensor is 0 then the output of the protocol is 0, otherwise the output of the protocol is 1. The complexity of the protocol is the number of communication rounds. We say the protocol nondeterministically computes f if the probability that the output equals 1 is nonzero if $f(x_1, \ldots, x_k) = 1$ and the probability that the output equals 0 is one if $f(x_1, \ldots, x_k) = 0$.

In particular, let $\mathrm{NQ}'(f)$ denote the complexity of the function f in the broadcast channel model. Then $\mathrm{NQ}'(f) \leq \mathrm{NQ}(f) + 1$. Indeed, consider a protocol in the NQ-model that computes f using $|\mathrm{GHZ}_r^k\rangle$ as a starting state. To simulate this protocol in the NQ'-model, one of the players uses the broadcast channel $\lceil \log_2 r \rceil$ times to create $|\mathrm{GHZ}_r^k\rangle$. Then the players proceed with the local quantum operations to compute f. This finishes the proof. We don't know whether the inequality $\mathrm{NQ}(f) \leq \mathrm{NQ}'(f)$ holds.

3 Nondeterministic log-rank conjecture for message-passing protocols

▶ **Definition 7.** Let $\mathrm{NQ}_0(f)$ be the minimal complexity of a protocol that nondeterministically computes f, without preshared entanglement (that is, the space is initialised in the state $|x_1 \cdots x_k\rangle$ instead of $|x_1 \cdots x_k\rangle |\mathrm{GHZ}_r^k\rangle$) but with the added ability for every player to send a qubit to another player. This communication happens in communication rounds; the protocol specifies per round who communicates to whom, independently of the input. The complexity of such a protocol is the total number rounds.

Villagra et al. [19] show that $\mathrm{NQ}_0(f)$ is at least the logarithm of the support rank of f. They furthermore conjecture that $\mathrm{NQ}_0(f)$ is upper bounded by a polynomial in the logarithm of the support rank. The following theorem proves this conjecture.

▶ **Theorem 8** ("Nondeterministic log-rank conjecture"). *Let $f : [2^n]^k \to \{0,1\}$. Then we have $\mathrm{NQ}(f) \leq \mathrm{NQ}_0(f) \leq (k-1)\,\mathrm{NQ}(f)$.*

Proof. For the first inequality, suppose we have an NQ_0-protocol for f. We replace the communication of a qubit by the nondeterministic teleportation of that qubit. Beforehand, all players agree on the basis in which the teleportation should happen. If any teleportation during the protocol does not happen in this basis, then the player that notices this sets his output register R_i to $|0\rangle$.

For the second inequality, suppose we have an NQ-protocol for f which uses a GHZ-state of rank r. Then we can construct a NQ_0-protocol for f as follows. The players start with no shared entanglement. Player 1 constructs a GHZ-state of rank r locally. In the first $k-1$ communication rounds, player 1 distributes the GHZ-state over the other $k-1$ players. After that, the players perform the NQ-protocol. The resulting NQ_0-protocol has complexity at most $(k-1)\,\mathrm{NQ}(f)$. ◀

To say something about the 'tightness' of Theorem 8 we consider the natural easy function in the NQ-model, namely $f(x_1, \ldots, x_k) = [x_1 = x_2 = \cdots = x_k]$ with $x_i \in [2^n]$.

▶ **Proposition 9** (Single bit inputs). *Let $f : [2]^k \to \{0,1\}$ be the function defined by $f(x_1, \ldots, x_k) = [x_1 = x_2 = \cdots = x_k]$ for $x_i \in [2]$. Then we have $\mathrm{NQ}(f) = 1$ and $\mathrm{NQ}_0(f) = (k-1)\,\mathrm{NQ}(f)$.*

Proof. Note that the tensor of this function is GHZ_2^k, so $\mathrm{NQ}(f) = 1$. Now consider a protocol that nondeterministically computes f without preshared entanglement and r rounds of communication. We may assume, without loss of generality, that the protocol consists of a first phase in which the players communicate and a second phase in which the players only do local quantum operations. After the first phase the players are sharing some state E consisting of EPR-pairs shared among certain pairs of the players. We thus obtain a local linear map which maps $\sum_x |x\rangle\, E$ to a tensor with the same support as GHZ_2^k. However, if $r < k-1$, then, viewing E as a graph, E is disconnected. Therefore there is a grouping of the players into two groups such that there are no EPR-pairs between the groups. Such a state cannot be converted to a GHZ_2^k state by SLOCC. ◀

Asymptotically, we can improve the relationship stated in Theorem 8, as follows.

▶ **Theorem 10** (Asymptotic upper bound). *For any $\varepsilon > 0$, there is an n_0 such that for all $f : [m]^k \to \{0,1\}$, if $\mathrm{NQ}(f) > n_0$, then*

$$\mathrm{NQ}_0(f) \leq \frac{(k+\varepsilon)}{2}\,\mathrm{NQ}(f).$$

To prove Theorem 10 we use the theory of asymptotic SLOCC conversion rates.

▶ **Definition 11.** Given tensors $\psi \in V_1 \otimes \cdots \otimes V_k$ and $\phi \in W_1 \otimes \cdots \otimes W_k$, we say that ψ can be transformed into ϕ via SLOCC operations, if there exist linear transformations $A_i : V_i \to W_i$ such that $\phi = (A_1 \otimes \cdots \otimes A_k)\psi$; and we write $\psi \xrightarrow{\text{SLOCC}} \phi$. Define

$$\omega_n(\psi,\phi) = \frac{1}{n}\inf\{m \in \mathbf{N}_{\geq 1} \mid \psi^{\otimes m} \xrightarrow{\text{SLOCC}} \phi^{\otimes n}\}$$

and

$$\omega(\psi,\phi) = \lim_{n\to\infty} \omega_n(\psi,\phi).$$

▶ **Lemma 12.** *The limit $\omega(\psi,\phi)$ exists and for all n the inequality $\omega_n(\psi,\phi) \geq \omega(\psi,\phi)$ holds; in other words, $\omega_n = \omega + o(1)$.*

▶ **Theorem 13** (Vrana-Christandl [20]). *Let $\mathrm{GHZ}_2^{K_k}$ be the k-party tensor consisting of EPR-pairs between any parties. Then*

$$\omega(\mathrm{GHZ}_2^{K_k}, \mathrm{GHZ}_2^k) = \frac{1}{k-1}.$$

In other words, for any $\varepsilon > 0$, there is an n_0 such that for all $n > n_0$,

$$(\mathrm{GHZ}_2^{K_k})^{\otimes n(\frac{1}{k-1}+\varepsilon)} \xrightarrow{\text{SLOCC}} (\mathrm{GHZ}_2^k)^{\otimes n}.$$

Proof of Theorem 10. Creating $\mathrm{GHZ}_2^{K_k}$ in the NQ_0-model costs $\binom{k}{2}$ messages. Asymptotically, we can transform $1/(k-1)$ copies of $\mathrm{GHZ}_2^{K_k}$ to one copy of GHZ_2^k by SLOCC. More precisely, by Theorem 13, for any $\varepsilon > 0$, there is an n_0 such that for all $n > n_0$,

$$(\mathrm{GHZ}_2^{K_k})^{\otimes \frac{n}{k-1}+\varepsilon n} \xrightarrow{\text{SLOCC}} (\mathrm{GHZ}_2^k)^{\otimes n}.$$

We conclude that, for any $\varepsilon > 0$, there is an n_0 such that for all $n > n_0$, $\binom{k}{2}(\frac{n}{k-1} + \varepsilon n) = ((k+\varepsilon')n)/2$ messages are sufficient to generate $(\mathrm{GHZ}_2^k)^{\otimes n}$ by SLOCC.

To prove the theorem, suppose we have an NQ-protocol for f which uses a GHZ state of rank 2^n and no communication. Consider the following NQ_0-protocol for f. Create a GHZ-state of rank 2^n by sending $\frac{(k+\varepsilon')n}{2}$ messages and then continue with the NQ-protocol. ◀

The following proposition says that the asymptotic relationship of Theorem 10 is tight.

▶ **Proposition 14** (n-bit inputs). *Let $f : [2^n]^k \to \{0,1\}$ be the function defined by $f(x_1,\ldots,x_k) = [x_1 = x_2 = \cdots = x_k]$ for $x_i \in [2^n]$. Then we have $\mathrm{NQ}(f) = n$ and $\mathrm{NQ}_0(f) \geq \frac{k}{2}\,\mathrm{NQ}(f)$.*

Proof. As in the previous proof, note that the tensor corresponding to f is $\mathrm{GHZ}_{2^n}^k$. Suppose there is an NQ_0 protocol using r messages. View the communication pattern of this protocol as an undirected multigraph G (i.e. parallel edges are allowed) on k vertices. Note that G has r edges. Let $E = \mathrm{GHZ}_2^G$ be the tensor that has an EPR pair at every edge in G. The protocol

yields an SLOCC transformation of E to $\mathrm{GHZ}_{2^n}^k$. Let ℓ be the minimal number of edges across any cut of G. Then ℓ is at most the minimal degree d of G. The sum of all degrees in G equals $2r$, so $k\ell \leq kd \leq 2r$, which implies the inequality $r \geq k\ell/2$. The number ℓ is equal to $\min_{S\subseteq[k]} \log_2 \mathrm{rk}_S(E)$, where $\mathrm{rk}_S(E)$ denotes the rank of E after flattening according to the set S. This value cannot increase under any SLOCC transformation. Now note that $\log_2 \mathrm{rk}_{\{i\}}(\mathrm{GHZ}_{2^n}^k) = n$ for any $i \in [k]$, so $\ell \geq n$. We conclude that $r \geq kn/2$. ◀

▶ Remark. Another way to prove Proposition 14 is to first symmetrize the protocol to obtain an SLOCC transformation of a state E with $\log_2 \mathrm{rk}_{\{i\}}(E) = (k-1)!2r$ to the state $\mathrm{GHZ}_{2^{k!n}}^k$. We have $\log_2 \mathrm{rk}_{\{i\}}(\mathrm{GHZ}_{2^{k!n}}^k) = k!\,n$. Since $\log_2 \mathrm{rk}_{\{i\}}$ is an SLOCC-monotone, we obtain the inequality $(k-1)!\,2r \geq k!\,n$ and hence $r \geq kn/2$.

4 Cyclic equality problem

The two-player equality problem EQ_n is the problem of Alice and Bob having to decide whether their n-bit inputs are equal. Since the identity matrix has full support rank, we have $\mathrm{NQ}(\mathrm{EQ}_n) = n$. We generalize EQ_n to multiple players as follows. Let G be an undirected graph. Let EQ_n^G be the problem of $|G|$ players having to solve the n-bit equality problem between players connected by edges. (Note that this definition naturally generalizes to hypergraphs.) If G is a bipartite graph, one easily sees that by grouping the players we can transform the problem into an equality problem on en bits EQ_{en}, where e is the number of edges in the graph. Therefore $\mathrm{NQ}(\mathrm{EQ}_n^G) = en$, that is, the trivial protocol is optimal for bipartite graphs. On the other hand, if G contains an odd cycle, then this argument fails. In the rest of this paper we will focus on the extreme case of G being an odd cycle and investigate the complexity of the corresponding equality problem.

▶ **Definition 15.** The k-player *cyclic equality problem* on n bits $\mathrm{EQ}_n^{C_k}$ is the function

$$\mathrm{EQ}_n^{C_k} : ([2^n]\times[2^n])^k \to \{0,1\} : (a_1b_1,\ldots,a_kb_k) \mapsto \begin{cases} 1 & \text{if } b_1 = a_2,\ b_2 = a_3,\ \ldots,\ b_k = a_1 \\ 0 & \text{otherwise,} \end{cases}$$

that is, the players are arranged in a circle; player i receives two n-bit inputs a_i, b_i and has to decide whether $a_i = b_{i-1}$ and $b_i = a_{i+1}$, where the indices are taken modulo k.

It turns out that the tensor corresponding to this function is a generalisation of the *matrix multiplication tensor*, one of the central objects of study in algebraic complexity theory. This tensor arises as follows in algebraic complexity theory. Consider the bilinear map

$$\mathbf{C}^{m\times m} \times \mathbf{C}^{m\times m} \to \mathbf{C}^{m\times m} : (A,B) \mapsto AB$$

which multiplies two complex $m \times m$ matrices. Any bilinear map $U \times V \to W$ corresponds canonically to a tensor in $U \otimes V \otimes W$. The number of multiplications in the field $\mathbf{C}$ necessary to perform the bilinear map is equal to the tensor rank of the corresponding tensor, up to a factor 2. The tensor corresponding to the matrix multiplication map is

$$\langle m,m,m\rangle \coloneqq \sum_{x\in[m]^3} |x_1x_2\rangle|x_2x_3\rangle|x_3x_1\rangle\,.$$

A natural generalisation of the tensor $\langle m,m,m\rangle$ to a k-party tensor is the so-called *iterated matrix multiplication* tensor

$$\mathrm{IMM}_m^k \coloneqq \sum_{x\in[m]^k} |x_1x_2\rangle|x_2x_3\rangle\cdots|x_kx_1\rangle\,.$$

Clearly, $\mathrm{IMM}_m^3 = \langle m, m, m\rangle$. The tensor IMM_m^k corresponds to the multilinear map

$$(\mathbf{C}^{m\times m})^{\times k} \to \mathbf{C} : (A_1, A_2, \ldots, A_k) \mapsto \mathrm{tr}(A_1 A_2 \cdots A_k)$$

which computes the trace of the product of k matrices. We note that, when viewed as a polynomial in the matrix entries, IMM_m^k plays a special role in the field of arithmetic circuits and geometric complexity theory. Namely, IMM_3^k is complete for the class $\mathrm{VP_e}$ of families of polynomials computable by small formulas [2], and IMM_k^k is complete for the class VQP, for which the determinant is also complete [4]. The following connection between iterated matrix multiplication and cyclic equality is readily observed.

▶ **Proposition 16.** *The tensor corresponding to the cyclic equality function* $\mathrm{EQ}_n^{C_k}$ *on* n *bits is the iterated matrix multiplication tensor* $\mathrm{IMM}_{2^n}^k$ *with* $2^n \times 2^n$ *matrices. Therefore, we have the equalities* $\mathrm{NQ}(\mathrm{EQ}_n^{C_k}) = \log_2 \mathrm{R_s}(\mathrm{IMM}_{2^n}^k)$ *and* $\underline{\mathrm{NQ}}(\mathrm{EQ}_n^{C_k}) = \log_2 \underline{\mathrm{R}}_{\mathrm{s}}(\mathrm{IMM}_{2^n}^k)$

The remainder of this paper is organized as follows. In the following four paragraphs we do the following: (1) we show that in the classical model, the naïve protocol in which every player broadcasts his input is optimal; (2) we show that when k is even the naïve protocol is optimal quantumly; (3) we exhibit nontrivial protocols when $n = 1$ and $k = 3$ or $k = 5$; (4) we show nontrivial lower bounds on the quantum complexity by use of Young flattenings. Finally, in the last section, we show that the Strassen laser method yields nontrivial protocols for all odd $k \geq 3$, asymptotically.

Classical lower bound with the fooling set method. We will show that in the classical situation the trivial protocol is always optimal. To prove a lower bound on the classical complexity of the cyclic equality problem we use the fooling set method. This is a method from the 2-player setting that extends naturally to the k-player setting.

▶ **Theorem 17.** *The classical nondeterministic communication complexity* $\mathrm{N}(\mathrm{EQ}_n^{C_k})$ *of the cyclic equality problem equals* kn*.*

Proof. Let $S \subseteq [2^{2n}]^k$ be the set of 1-inputs of the function $\mathrm{EQ}_n^{C_k}$. This set has size 2^{kn}. Let Π be a classical protocol for $\mathrm{EQ}_n^{C_k}$ and denote by $\Pi_r(x_1, \ldots, x_k)$ the sequence of messages sent by the players in the protocol Π on input $x \in [2^{2n}]^k$ and private randomness $r \in [m]^k$. Suppose there are distinct 1-inputs $x, y \in S$ and private randomnesses $r, s \in [m]^k$ such that $\Pi_r(x_1, \ldots, x_k) = \Pi_s(y_1, \ldots, y_k)$. There is an i such that $x_i \neq y_i$, say $i = 1$. We have $\Pi_r(x_1, \ldots, x_k) = \Pi_{(r_1, s_2, \ldots, s_k)}(x_1, y_2, \ldots, y_k)$, so the protocol outputs 1 on input $x_1, y_2, \ldots, y_k$ with randomness $(r_1, s_2, \ldots, s_k)$. However, $x_1, y_2, \ldots, y_k$ is a 0-input, a contradiction. Therefore, $\Pi_r(x_1, \ldots, x_k) \neq \Pi_s(y_1, \ldots, y_k)$. We conclude that $\mathrm{N}(\mathrm{EQ}_n^{C_k}) \geq \log_2(|S|)$. ◀

An even number of quantum players. When k is even, the cycle graph C_k is bipartite, and, as mentioned above, the best protocol for an equality problem on a bipartite graph is the trivial protocol. We record this statement in terms of border support rank in the following proposition.

▶ **Proposition 18.** *For even* k*,* $m^k \leq \underline{\mathrm{R}}_{\mathrm{s}}(\mathrm{IMM}_m^k)$*. As a consequence, we have the equalities* $\underline{\mathrm{NQ}}(\mathrm{EQ}_n^{C_k}) = \mathrm{NQ}(\mathrm{EQ}_n^{C_k}) = kn$*.*

Proof. Let t be a tensor with the same support as $\mathrm{IMM}_m^k \in (\mathbf{C}^{m^2})^{\otimes k}$. Label the players with the numbers $1, 2, \ldots, k$. Group the *even* players together and group the *odd* players

together and flatten the tensor t accordingly into a matrix A in $(\mathbf{C}^{m^2})^{\otimes k/2} \otimes (\mathbf{C}^{m^2})^{\otimes k/2}$. The matrix A has the same support as the identity matrix in $(\mathbf{C}^{m^2})^{\otimes k/2} \otimes (\mathbf{C}^{m^2})^{\otimes k/2}$ and thus has rank m^k. ◀

Note that for odd k the above proof yields the lower bound $m^{k-1} \leq \underline{\mathrm{R}}_{\mathrm{s}}(\mathrm{IMM}_m^k)$. We will show in Theorem 20 that this lower bound is not tight.

Nontrivial 3-player and 5-player quantum protocols. In the 3-player situation, Strassen's celebrated decomposition of the tensor $\mathrm{IMM}_2^3 = \langle 2,2,2\rangle$ into a sum of 7 simple tensors [17] gives a nontrivial protocol for $\mathrm{EQ}_1^{C_3}$, and thus $\mathrm{NQ}(\mathrm{EQ}_1^{C_3}) \leq \log_2(7)$. We show that for 5 players there also exists a nontrivial protocol for $\mathrm{EQ}_1^{C_5}$, as follows. Recall that we have defined $\mathrm{IMM}_2^5 = \sum_{i\in[2]^5} |i_1i_2\rangle|i_2i_3\rangle|i_3i_4\rangle|i_4i_5\rangle|i_5i_1\rangle$. Observe that an upper bound $\mathrm{R}(\mathrm{IMM}_2^5) \leq r$ implies $\mathrm{R}(\mathrm{IMM}_n^5) \leq \mathcal{O}(n^{\log_2(r)})$ by taking tensor powers of IMM_2^5.

▶ **Theorem 19.** $\mathrm{R}(\mathrm{IMM}_2^5) \leq 31$, *and thus* $\mathrm{NQ}(\mathrm{EQ}_1^{C_5}) \leq \log_2(31)$.

Proof. Let $|-\rangle := |1\rangle - |2\rangle$, $|+\rangle := |1\rangle + |2\rangle$ and $|\Phi^+\rangle = |11\rangle + |22\rangle$. Let $\mathrm{Cyc}_5 := \sum_{\sigma\in C_5} \sigma$ be the cyclic symmetrizer acting on $(\mathbf{C}^4)^{\otimes 5}$ by permuting the 5 parties, and moreover let $\mathrm{Sym}_2 := \sum_{\sigma\in S_2} \sigma$ be a 'local symmetrizer' acting diagonally on $(\mathbf{C}^2)^{\otimes 10}$ by permuting the basis states $|1\rangle$ and $|2\rangle$ of each $\mathbf{C}^2$. Let

$$\begin{aligned} t := &- |-1\rangle\,|11\rangle\,|11\rangle\;|1+\rangle\,|22\rangle \\ &- |-1\rangle\,|12\rangle\,|21\rangle\;|1+\rangle\,|22\rangle \\ &- |\Phi^+\rangle\,|22\rangle\,|-1\rangle|1+\rangle\,|22\rangle. \end{aligned}$$

By direct computation, we see that $\mathrm{IMM}_2^5 = \mathrm{Cyc}_5\big(\mathrm{Sym}_2(t)\big) + |\Phi^+\rangle^{\otimes 5}$. We observe that the right hand side yields a sum of 31 simple tensors. ◀

We have a proof generalizing Theorem 19 to $\mathrm{R}(\mathrm{IMM}_2^k) \leq 2^k - 1$ for all odd k, which will appear in a forthcoming paper [8].

Quantum lower bound with Young flattenings. Let $t \in V_1 \otimes V_2 \otimes V_3$ be some 3-tensor. By grouping V_1 and V_2, the tensor t can be viewed as a matrix $A \in (V_1 \otimes V_2) \otimes V_3$; this is called a *flattening*. The rank of the flattening A is a lower bound for the border rank of t and thus we obtain lower bounds on the border rank of tensors by computing the rank of their flattenings. However, this type of lower bound can never be bigger than the dimension of any local space V_i, and there do exist tensors with border rank larger than the local dimensions, for example the matrix multiplication tensor $\langle 2,2,2\rangle$.

One approach to overcome this 'local dimension limitation' is as follows. We let $\phi : V_2 \to W_1 \otimes W_2$ be a linear map such that $\mathrm{R}(\phi(v)) \leq e$ for all $v \in V_2$. By applying ϕ to the central tensor leg of t, we transform t into a 4-tensor $s \in V_1 \otimes W_1 \otimes W_2 \otimes V_3$. Next, we flatten s to a matrix $A \in (V_1 \otimes W_1) \otimes (W_2 \otimes V_3)$. The rank of A divided by e is a lower bound for the border rank of t. We will be using a specific linear map ϕ which originates from the representation theory of the general linear group. When one takes such representation theoretic maps ϕ to construct a flattening as above one speaks of a *Young flattening* [13]. An early appearance of this type of flattening can be recognized in the work of Strassen [18]. The following lower bound is obtained with a Young flattening.

▶ **Theorem 20.** *For odd* $k \geq 3$, $(2n^2 - n)n^{k-3} \leq \underline{\mathrm{R}}_{\mathrm{s}}(\mathrm{IMM}_n^k)$. *As a consequence, we have the lower bound* $(k-1)n + \log_2(2 - \frac{1}{n}) \leq \underline{\mathrm{NQ}}(\mathrm{EQ}_n^{C_k})$.

Proof. Let $k = 3$. The proof for odd $k > 3$ goes similarly after having grouped the k parties appropriately to 3 parties. For a vector space V, let $\wedge^a V$ be the ath exterior power of V. Define the linear map

$$\begin{aligned}\phi : \mathbf{C}^{2n-1} &\to \wedge^p \mathbf{C}^{2n-1} \otimes \wedge^{p+1} \mathbf{C}^{2n-1} \\ |j\rangle &\mapsto \sum_{j_1<\cdots<j_p} |j_1\rangle\wedge\cdots\wedge|j_p\rangle \otimes |j_1\rangle\wedge\cdots\wedge|j_p\rangle\wedge|j\rangle\,,\end{aligned}$$

and note that the rank of the matrix $\phi(v)$ equals $\binom{2n-2}{p}$ for any $v \in \mathbf{C}^{2n-1}$. We consider the tensor

$$t_1 \coloneqq \sum_i \alpha_{i_1,i_2,i_3} |i_1 i_2\rangle |i_2 i_3\rangle |i_3 i_1\rangle \in \mathbf{C}^{n^2} \otimes \mathbf{C}^{n^2} \otimes \mathbf{C}^{n^2},$$

where i runs over $[n]^3$ and the α_{i_1,i_2,i_3} are nonzero complex numbers. The border rank of t_1 is at least the border rank of

$$t_2 \coloneqq \sum_i \alpha_{i_1,i_2,i_3} |i_1 i_2\rangle |i_2 + i_3 - 1\rangle |i_3 i_1\rangle \in \mathbf{C}^{n^2} \otimes \mathbf{C}^{2n-1} \otimes \mathbf{C}^{n^2}.$$

Apply ϕ to the central tensor leg of t_2 and then flatten to obtain

$$A \coloneqq \sum_i \sum_{j_1<\cdots<j_p} \alpha_{i_1,i_2,i_3} |i_1 i_2\rangle |j_1\rangle\wedge\cdots\wedge|j_p\rangle \otimes |j_1\rangle\wedge\cdots\wedge|j_p\rangle\wedge|i_2 + i_3 - 1\rangle |i_3 i_1\rangle\,.$$

View A as a direct sum of n matrices $A_{i_1} \in (\mathbf{C}^n \otimes \wedge^p \mathbf{C}^{2n-1}) \otimes (\wedge^{p+1}\mathbf{C}^{2n-1} \otimes \mathbf{C}^n)$; the matrix A_{i_1} corresponds to the linear map

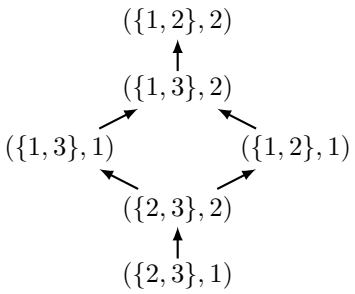
$$f_{i_1} : |i_2\rangle |j_1\rangle\wedge\cdots\wedge|j_p\rangle \mapsto \sum_{i_3} \alpha_{i_1,i_2,i_3} |j_1\rangle\wedge\cdots\wedge|j_p\rangle\wedge|i_2 + i_3 - 1\rangle |i_3\rangle\,.$$

Let $p = n - 1$. We claim that every matrix A_{i_1} is upper triangular with elements α_{i_1,i_2,i_3} on the diagonal, up to permutations of the rows and columns. Assuming the claim is true, we get that $\mathrm{R}(A) = \sum_{i_1} \mathrm{R}(A_{i_1}) = n \dim(\mathbf{C}^n \otimes \wedge^{n-1}\mathbf{C}^{2n-1}) = n^2\binom{2n-1}{n-1}$. This implies the lower bound $\underline{\mathrm{R}}_{\mathrm{s}}(\mathrm{IMM}_n^3) \geq n^2\binom{2n-1}{n-1}/\binom{2n-2}{n-1} = 2n^2 - n$.

To prove this claim we define a partial order on the basis elements $|j_1\rangle \wedge \cdots \wedge |j_n\rangle \otimes |\ell\rangle$ of the target space of A_{i_1}. We will use the same partial order as Landsberg and Michałek [12]. Denote the basis elements of the target space by (P, ℓ) with P an n-subset of $[2n-1]$ and $\ell \in [n]$. Let (P_1, ℓ_1) and (P_2, ℓ_2) be two such basis elements and define $\ell \coloneqq \min(\ell_1, \ell_2)$. We say $(P_1, \ell_1) < (P_2, \ell_2)$

1. if the ordered sequence of the ℓ smallest elements in P_2 is lexicographically smaller than the ordered sequence of the ℓ smallest elements in P_1;
2. or if the sequences of ℓ smallest elements are equal and $\ell_1 < \ell_2$.

One checks that this defines a partial order and that the unique minimal element in this order is $(\{n, \ldots, 2n-1\}, 1)$. For example, with $n = 2$ the partial order has the following Hasse diagram.



We prove the claim by induction on $<$, with the minimal element as a base case. For now let all the α_{i_1,i_2,i_3} be 1. First, under A_{i_1} we have

$$|n\rangle \otimes |n+1\rangle \wedge \cdots \wedge |2n-1\rangle \mapsto |n\rangle \wedge \cdots \wedge |2n-1\rangle \otimes |1\rangle\,,$$

so the minimal element $(\{n,\dots,2n-1\},1)$ is in the image of A_{i_1}. Let (P,ℓ) be in the target space of A_{i_1} and assume that every (P',ℓ') with $(P',\ell') < (P,\ell)$ is in the image. Write $P=(p_1,\dots,p_n)$ with $p_1 \le \cdots \le p_n$. Under A_{i_1} we have,

$$|p_1\rangle \wedge \cdots \widehat{p_\ell} \cdots \wedge |p_n\rangle \otimes |1+p_\ell-\ell\rangle \mapsto \sum_m |p_1\rangle \wedge \cdots \widehat{p_\ell} \cdots \wedge |p_n\rangle \wedge |p_\ell-\ell+m\rangle \otimes |m\rangle\,.$$

Taking $m=\ell$, one sees that the basis element (P,ℓ) is present in the sum. Moreover, for any other (P',m) appearing in the sum we have $(P',m) < (P,\ell)$. Indeed, if $m>\ell$, then $p_\ell-\ell+m > p_\ell$, so the smallest ℓ elements in P' are lexicographically larger than the smallest ℓ elements in P, meaning $(P',m)<(P,\ell)$ by rule 1; if $m<\ell$, then $p_m \le p_\ell-\ell+m < p_\ell$, so the m smallest elements of P' and P are equal, meaning $(P',m)<(P,\ell)$ by rule 2. Therefore, the basis element (P,ℓ) is in the image. This argument shows that A_{i_1} has full rank. Moreover, this argument shows that, up to a permutation of the rows and columns, the matrix A_{i_1} is upper triangular with ones on the diagonal. Repeating this argument with general values for α_{i_1,i_2,i_3} proves the claim. ◀

▶ Remark. The lower bound in Theorem 20 improves a lower bound of Ikenmeyer on the border support rank of IMM_n^3 [11, 8.2.17].

5 Strassen's laser method for iterated matrix multiplication

In this section we show that $\mathrm{NQ}(\mathrm{EQ}_n^{C_k}) < kn$ for odd k. We will prove this result in the language of algebraic complexity theory.

▶ **Definition 21.** Define $\omega_k := \inf\{\alpha \in \mathbf{R} \mid \mathrm{R}(\mathrm{IMM}_n^k) \in \mathcal{O}(n^\alpha)\}$. We call this the *exponent* of iterated matrix multiplication. Define $\omega_{\mathrm{s},k} := \inf\{\alpha \in \mathbf{R} \mid \mathrm{R_s}(\mathrm{IMM}_n^k) \in \mathcal{O}(n^\alpha)\}$. We call this the *support rank exponent* of iterated matrix multiplication.

Asymptotically, we have $\mathrm{NQ}(\mathrm{EQ}_n^{C_k}) \le \omega_{\mathrm{s},k}\, n + \mathcal{O}(1) \le \omega_k\, n + \mathcal{O}(1)$. The exponents ω_3 and $\omega_{\mathrm{s},3}$ are known as ω and ω_{s} in the literature. The support rank exponent of matrix multiplication was first studied by Cohn and Umans [9]. The best upper bound on ω_s comes from the upper bound $\omega \le 2.3728639$ of Le Gall [14]. Interestingly, Cohn and Umans show the relationship

$$\omega \le (3\omega_{\mathrm{s}} - 2)/2.$$

Therefore, one way of finding upper bounds on ω is to construct an efficient quantum communication protocol for the cyclic equality problem $\mathrm{EQ}_n^{C_3}$. On the other hand, this observation indicates that improving the bounds on the quantum communication complexity of $\mathrm{EQ}_n^{C_k}$ is a hard problem.

For any k we have $k-1 \le \omega_k \le k$, and if k is even, then $\omega_k = k$ (Proposition 18). The aim of this section will be to show: if $k \ge 3$ is odd, then

$$\omega_k < k.$$

Schönhage τ-theorem. In this section we will generalize some tools for obtaining upper bounds on the exponent of ω_3 to all exponents ω_k, in particular, we generalize the Schönhage τ-theorem. The proofs in this section are straightforward generalizations of the proofs for $k = 3$ which can be found in [5]. In the next paragraph, we will use Strassen's laser method to show that $\omega_k < k$ for all odd k.

First we recall an important relationship between border rank and rank. We use the following more precise notion of border rank. Let $h \in \mathbf{N}$ and let t be a tensor in $\mathbf{C}^{\otimes m_1} \otimes \cdots \otimes \mathbf{C}^{\otimes m_k}$. Define $\mathrm{R}_h(t)$ to be the minimum number r such that there exist vectors $v_i^j \in (\mathbf{C}[\varepsilon])^{m_j}$ that satisfy $\sum_{i=1}^r v_i^1 \otimes \cdots \otimes v_i^k = \varepsilon^h t + \mathcal{O}(\varepsilon^{h+1})$. A well-known but nontrivial result is that $\underline{\mathrm{R}}(t) = \min_h \mathrm{R}_h(t)$. It is not hard to show that $\mathrm{R}_{h+h'}(t \otimes t') \leq \mathrm{R}_h(t)\,\mathrm{R}_{h'}(t')$. The relationship we are talking about is the following.

▶ **Proposition 22.** *For every $h, k \in \mathbf{N}$, there is a number c_h such that for all tensors $t \in \mathbf{C}^{m_1} \otimes \cdots \otimes \mathbf{C}^{m_k}$, $\mathrm{R}(t) \leq c_h\,\mathrm{R}_h(t)$. The number c_h depends polynomially on h.*

Proof. Let t be a tensor in $\mathbf{C}^{m_1} \otimes \cdots \otimes \mathbf{C}^{m_k}$ with $\mathrm{R}_h(t) = r$. Then there are $v_i^j \in (\mathbf{C}[\varepsilon])^{m_j}$ such that

$$\sum_{i=1}^{r} v_i^1 \otimes \cdots \otimes v_i^k = \varepsilon^h t + \mathcal{O}(\varepsilon^{h+1}).$$

Decomposing every v_i^j into ε-homogeneous components $v_i^j = \sum_{a_j=0}^{h} \varepsilon^{a_j} v_i^j(a_j)$, and collecting powers of ε gives

$$\sum_{i=1}^{r} \sum_{a_1,\ldots,a_k \in [h]} \varepsilon^{a_1+\cdots+a_k}\, v_i^1(a_1) \otimes \cdots \otimes v_i^k(a_k) = \varepsilon^h t + \mathcal{O}(\varepsilon^{h+1}).$$

Taking only the summands such that $a_1 + \cdots + a_k = h$ gives a rank decomposition of t. There are $\binom{h+k-1}{k-1} r$ such summands. ◀

Next, we show that an upper bound on the border rank of 'unbalanced' iterated matrix multiplication tensors yields and upper bound on ω_k. Define the tensor $\langle n_1, n_2, \ldots, n_k \rangle$ to be

$$\sum_{x \in [n_1] \times \cdots \times [n_k]} |x_1 x_2\rangle\, |x_2 x_3\rangle \cdots |x_k x_1\rangle\,.$$

So $\mathrm{IMM}_n^k = \langle n, n, \ldots, n \rangle$ (n occurs k times).

▶ **Proposition 23.** *If $\underline{\mathrm{R}}(\langle n_1, n_2, \ldots, n_k \rangle) \leq r$, then $\omega_k \leq k \log_{n_1 \cdots n_k} r$.*

Proof. Let $N = n_1 \cdots n_k$. There is an h such that $\mathrm{R}_h(\langle n_1, \ldots, n_k \rangle) \leq r$. By taking the tensor product of all cyclic shifts of $\langle n_1, \ldots, n_k \rangle$, we get $\mathrm{R}_{kh}(\langle N, \ldots, N \rangle) \leq r^k$ and thus $\mathrm{R}_{khs}(\langle N^s, \ldots, N^s \rangle) \leq r^{ks}$ for all s. Hence $\mathrm{R}(\langle N^s, \ldots, N^s \rangle) \leq c_{khs} r^{ks}$ for some number c_{khs} which is constant in N. Therefore,

$$\omega \leq \log_{N^s}(c_{khs} r^{ks}) = ks \log_{N^s}(r) + \log_{N^s}(c_{khs}).$$

If s goes to infinity then $\log_{N^s}(c_{khs})$ goes to zero, so $\omega_k \leq k \log_N(r)$. ◀

The real workhorse is the following straightforward generalization of a theorem of Schönhage [16].

▶ **Proposition 24** (k-party Schönhage τ-theorem). *Suppose that $r > p$ and*

$$\underline{\mathrm{R}}\Big(\bigoplus_{i=1}^{p}\langle n_1^i, n_2^i, \ldots, n_k^i\rangle\Big) \leq r.$$

Define τ by $\sum_{i=1}^{p}\big(\prod_{j=1}^{k} n_j^i\big)^{\tau} = r$. Then $\omega_k \leq k\tau$

We follow the proof of [5]. We first prove two lemmas. For tensors $s, t \in \mathbf{C}^{m_1}\otimes\cdots\otimes\mathbf{C}^{m_k}$, let $s \leq t$ denote the existence of an SLOCC transformation mapping t to s. Let $a, b \in \mathbf{N}+1$.

▶ **Lemma 25.** *Let t be a tensor such that $\mathrm{R}(t^{\oplus a}) \leq b$. Then for all s, $\mathrm{R}((t^{\otimes s})^{\oplus a}) \leq \lceil b/a\rceil^s a$.*

Proof. We prove the lemma by induction over s. The base case $s = 1$ follows from the assumption. For the induction step, we have

$$(t^{\otimes s+1})^{\oplus a} = t^{\oplus a}\otimes t^{\otimes s} \leq \mathrm{GHZ}_b \otimes t^{\otimes s} = (t^{\otimes s})^{\oplus b},$$

and thus, by the induction hypothesis,

$$\mathrm{R}((t^{\otimes s+1})^{\oplus a}) \leq \mathrm{R}((t^{\otimes s})^{\oplus b}) \leq \mathrm{R}((t^{\otimes s})^{\oplus(\lceil b/a\rceil a)}) \leq \lceil\tfrac{b}{a}\rceil\lceil\tfrac{b}{a}\rceil^s a \leq \lceil\tfrac{b}{a}\rceil^{s+1} a,$$

proving the lemma. ◀

▶ **Lemma 26.** *If $\mathrm{R}(\langle n_1, n_2, \ldots, n_k\rangle^{\oplus a}) \leq b$, then $\omega_k \leq k\log_{n_1\cdots n_k}\lceil b/a\rceil$.*

Proof. The inequality $\mathrm{R}(\langle n_1, n_2, \ldots, n_k\rangle^{\oplus a}) \leq b$ implies by Theorem 25 the inequality $\mathrm{R}(\langle n_1^s, n_2^s, \ldots, n_k^s\rangle^{\oplus a}) \leq \lceil b/a\rceil^s a$ which by Proposition 23 yields

$$\omega_k \leq k\frac{s\log\lceil\frac{b}{a}\rceil + \log(a)}{s\log(n_1\cdots n_k)},$$

which goes to $k\log\lceil b/a\rceil/\log(n_1\cdots n_k)$ when s goes to infinity. ◀

Proof of Proposition 24. We assume $\underline{\mathrm{R}}\big(\bigoplus_{i=1}^{p}\langle n_1^i, n_2^i, \ldots, n_k^i\rangle\big) \leq r$. This implies that there is an $h \in \mathbf{N}$ such that $\mathrm{R}_h\big(\bigoplus_{i=1}^{p}\langle n_1^i, n_2^i, \ldots, n_k^i\rangle\big) \leq r$. Taking the sth tensor power gives $\mathrm{R}_{hs}\big((\bigoplus_{i=1}^{p}\langle n_1^i, n_2^i, \ldots, n_k^i\rangle)^{\otimes s}\big) \leq r^s$. We expand the tensor power to get

$$\mathrm{R}_{hs}\Big(\bigoplus_{\sigma}\Big(\bigotimes_{i=1}^{p}\langle (n_1^i)^{\sigma_i}, (n_2^i)^{\sigma_i}, \ldots, (n_k^i)^{\sigma_i}\rangle\Big)^{\oplus\binom{s}{\sigma_1,\ldots,\sigma_p}}\Big) \leq r^s,$$

where the first direct sum is over all p-tuples σ of nonnegative integers with sum s. We can also write this inequality as

$$\mathrm{R}_{hs}\Big(\bigoplus_{\sigma}\langle \textstyle\prod_i (n_1^i)^{\sigma_i}, \ldots, \prod_i (n_k^i)^{\sigma_i}\rangle^{\oplus\binom{s}{\sigma_1,\ldots,\sigma_p}}\Big) \leq r^s.$$

There exists a number c_{hs} depending polynomially on h and s such that

$$\mathrm{R}\Big(\bigoplus_{\sigma}\langle \textstyle\prod_i (n_1^i)^{\sigma_i}, \ldots, \prod_i (n_k^i)^{\sigma_i}\rangle^{\oplus\binom{s}{\sigma_1,\ldots,\sigma_p}}\Big) \leq c_{hs}\, r^s.$$

Define τ by $\sum_{i=1}^{p}\big(\prod_{j=1}^{k} n_j^i\big)^{\tau} = r$. Then $\sum_{\sigma}\binom{s}{\sigma_1,\ldots,\sigma_p}\big(\prod_i (n_1^i)^{\sigma_i}\cdots\prod_i (n_k^i)^{\sigma_i}\big)^{\tau} = r^s$. In this sum, consider the maximum summand and fix the corresponding σ. Define the numbers $n_j \coloneqq \prod_i (n_j^i)^{\sigma_i}$. Let $a \coloneqq \binom{s}{\sigma_1,\ldots,\sigma_p}$ and $b \coloneqq r^s c_{hs}$. We apply Theorem 26 to the inequality $\mathrm{R}(\langle n_1, \ldots, n_k\rangle^{\oplus a}) \leq b$ to obtain

$$\omega_k \leq k\tau + \frac{(p-1)\log(s+1) + \log(c_{hs})}{\log(n_1\cdots n_k)},$$

which goes to $k\tau$ when s goes to infinity. (See [5] for more details.) ◀

Strassen's laser method. We will now use Strassen's laser method to prove the main result of this section.

▶ **Theorem 27.** *For any odd k we have $\omega_k < k$.*

We will give a proof for the case $k = 5$, the other cases being similar. Define the 5-tensor $\mathrm{Str}_q^5 = \sum_{i=1}^q |ii000\rangle + |0ii00\rangle$ in $\mathbf{C}^{q+1} \otimes \mathbf{C}^q \otimes \mathbf{C}^{q+1} \otimes \mathbf{C} \otimes \mathbf{C}$.

▶ **Proposition 28.** $\underline{\mathrm{R}}(\mathrm{Str}_q^5) \leq q + 1$.

Proof. Expanding $\sum_{i=1}^q (|0\rangle + \varepsilon\,|i\rangle)\,|i\rangle\,(|0\rangle + \varepsilon\,|i\rangle)\,|0\rangle|0\rangle$ gives

$$\sum_{i=1}^{q} |0i000\rangle + \varepsilon \sum_{i=1}^{q} |ii000\rangle + |0ii00\rangle + \mathcal{O}(\varepsilon^2).$$

Subtracting $|0\rangle\big(\sum_{i=1}^q |i\rangle\big)|000\rangle$ yields $\varepsilon\,\mathrm{Str}_q^5 + \mathcal{O}(\varepsilon^2)$. ◀

▶ **Proposition 29.** $\mathrm{GHZ}_2^5 \leq \langle 2,2,2,2,2\rangle$.

Proof. Let ϕ be the map $|ab\rangle \mapsto \delta_{[a=b]}\,|a\rangle$. Apply $\phi^{\otimes 5}$ to $\langle 2,2,2,2,2\rangle$. This yields one copy of $\mathrm{GHZ}_2^{[5]}$. ◀

▶ Remark. We mention that the subrank result of Proposition 29 can by improved asymptotically in the sense that $\omega(\langle 2,2,2,2,2\rangle, \mathrm{GHZ}^5) = 1/2$ [20]. Using this fact in the proof of Theorem 27 gives the slightly better upper bound $\omega_k \leq \log_q((q+1)^k/4)$.

For the proof of Theorem 27 we have to define the notion of the decomposition of the support of a tensor and the corresponding inner and outer structure of a tensor. Let $I_1, \ldots, I_k$ be finite sets. A *decomposition* $\mathcal{D}$ of $I_1 \times \cdots \times I_k$ is a collection of sets I_i^j such that

$$I_i = \bigsqcup_j I_i^j,$$

meaning that for every i, $\cap_j I_i^j = \emptyset$ and $\cup_j I_i^j = I_i$. Let t be a tensor in $\mathbf{C}^{m_1} \otimes \cdots \otimes \mathbf{C}^{m_k}$ and index the basis elements in this space by elements of $[m_1] \times \cdots \times [m_k]$. Let $\mathcal{D}$ be a decomposition of $[m_1] \times \cdots \times [m_k]$. We view $\mathcal{D}$ as a 'cut' of $[m_1] \times \cdots \times [m_k]$ into smaller product sets and thus as a 'cut' of t into smaller tensors. We define $t|_{I_1^{j_1}, I_2^{j_2}, \ldots, I_k^{j_k}}$ to be the restriction of t to the basis elements in $I_1^{j_1} \times I_2^{j_2} \times \ldots \times I_k^{j_k}$. These smaller tensors we think of as the 'inner structure' of t. We define the 'outer structure' of t with respect to $\mathcal{D}$ to be the tensor $t_{\mathcal{D}}$ indexed by sequences $(j_1, \ldots, j_k)$ such that $t_{\mathcal{D}}$ has a 1 at position $(j_1, \ldots, j_k)$ if t restricted to $I_1^{j_1} \times \cdots \times I_k^{j_k}$ is not the zero tensor, and a 0 otherwise.

Proof of Theorem 27. We will give a proof for the case $k = 5$, the other cases being similar. Define a block decomposition $\mathcal{D}$ of the support $I_1 \times \cdots \times I_5$ of Str_q^5 by

$$I_1 = \{0\} \cup \{1, \ldots, q\}$$
$$I_2 = \{1, \ldots, q\}$$
$$I_3 = \{0\} \cup \{1, \ldots, q\}$$
$$I_4 = \{0\}$$
$$I_5 = \{0\}.$$

We have the outer structure $(\mathrm{Str}_q^5)_{\mathcal{D}} = |11000\rangle + |01100\rangle \cong |10100\rangle + |00000\rangle$. Note that this is just an EPR pair between party 1 and 3. The inner structures are $\sum_{i=1}^q |ii000\rangle$ and

$\sum_{i=1}^{q} |0ii00\rangle$, which are also known as $\langle 1, q, 1, 1, 1\rangle$ and $\langle 1, 1, q, 1, 1\rangle$. Let Cyc_5 be the map $t \mapsto t \otimes \sigma t \otimes \sigma^2 t \otimes \sigma^3 t \otimes \sigma^4 t$ with $\sigma = (12345)$. Let $\hat{\mathcal{D}} = \mathrm{Cyc}_5 \mathcal{D}$ be the naturally corresponding decomposition. Then

$$\langle 2, 2, 2, 2, 2\rangle^{\otimes s} = (\mathrm{Cyc}_5 \operatorname{Str}_q^5)^{\otimes s}_{\hat{\mathcal{D}}^{\otimes s}} \quad \text{and} \quad \underline{\mathrm{R}}\big((\mathrm{Cyc}_5 \operatorname{Str}_q^5)^{\otimes s}\big) \leq (q+1)^{5s}. \tag{1}$$

Note how the first statement relies on 5 being odd.

The inner structure of $(\mathrm{Cyc}_5 \operatorname{Str}_q^5)^{\otimes s}_{\hat{\mathcal{D}}^{\otimes s}}$ consists of tensors from $I \coloneqq \{\langle n_1, n_2, n_3, n_4, n_5\rangle \mid n_1 \cdots n_5 = q^{5s}\}$. Combining equation (1) with Proposition 29 gives that there are 2^s elements $t_1, t_2, \ldots \in I$ such that

$$\underline{\mathrm{R}}(t_1 \oplus t_2 \oplus \cdots) \leq (q+1)^{5s}.$$

Now the τ-theorem says that if we define τ by

$$2^s (q^{5s})^{\tau} = (q+1)^{5s}$$

then $\omega_5 \leq 5\tau$. Therefore,

$$\omega_5 \leq 5\tau \leq \log_q \frac{(q+1)^5}{2}$$

which gives $\omega_5 \leq 4.84438$. In general, one gets $\omega_k \leq \log_q \frac{(q+1)^k}{2}$ which is strictly smaller than k for q large enough. ◀

Acknowledgements. We thank Peter Bürgisser, Péter Vrana, Florian Speelman and Teresa Piovesan for helpful discussions.

References

1. Andris Ambainis, Harry Buhrman, Yevgeniy Dodis, and Hein Röhrig. Multiparty quantum coin flipping. In *Computational Complexity, 2004. Proceedings. 19th IEEE Annual Conference on*, pages 250–259. IEEE, 2004. arXiv:quant-ph/0304112, doi:10.1109/CCC.2004.1313848.
2. Michael Ben-Or and Richard Cleve. Computing algebraic formulas using a constant number of registers. *SIAM Journal on Computing*, 21(1):54–58, 1992. doi:10.1137/0221006.
3. Amey Bhangale and Swastik Kopparty. The complexity of computing the minimum rank of a sign pattern matrix. *arXiv preprint arXiv:1503.04486*, 2015. arXiv:1503.04486.
4. Markus Bläser. Complete problems for Valiant's class of qp-computable families of polynomials. In *Computing and Combinatorics*, pages 1–10. Springer, 2001. doi:10.1007/3-540-44679-6_1.
5. Markus Bläser. Fast matrix multiplication. *Theory of Computing, Graduate Surveys*, 5:1–60, 2013. doi:10.4086/toc.gs.2013.005.
6. Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi. *Algebraic complexity theory*, volume 315 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1997. With the collaboration of Thomas Lickteig. doi:10.1007/978-3-662-03338-8.
7. Eric Chitambar, Runyao Duan, and Yaoyun Shi. Tripartite entanglement transformations and tensor rank. *Physical review letters*, 101(14):140502, 2008. arXiv:0805.2977, doi:10.1103/PhysRevLett.101.140502.
8. Matthias Christandl and Jeroen Zuiddam. Tensor surgery and tensor rank. *arXiv preprint arXiv:1606.04085*, 2016. arXiv:1606.04085.

9 Henry Cohn and Christopher Umans. Fast matrix multiplication using coherent configurations. In *Proceedings of the Twenty-fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'13, pages 1074–1086, Philadelphia, PA, USA, 2013. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=2627817.2627894, arXiv:1207.6528.

10 Fulvio Gesmundo. Geometric aspects of iterated matrix multiplication. *Journal of Algebra*, 461:42–64, 2016. arXiv:1512.00766, doi:10.1016/j.jalgebra.2016.04.028.

11 Christian Ikenmeyer. *Geometric complexity theory, tensor rank, and Littlewood-Richardson coefficients*. PhD thesis, Universität Paderborn, 2013. URL: http://nbn-resolving.de/urn:nbn:de:hbz:466:2-10472.

12 Joseph M. Landsberg and Mateusz Michałek. On the geometry of border rank algorithms for matrix multiplication and other tensors with symmetry. *arXiv preprint arXiv:1601.08229*, 2016. arXiv:1601.08229.

13 Joseph M. Landsberg and Giorgio Ottaviani. New lower bounds for the border rank of matrix multiplication. *Theory Comput.*, 11:285–298, 2015. arXiv:1112.6007, doi:10.4086/toc.2015.v011a011.

14 François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ISSAC'14, pages 296–303, New York, NY, USA, 2014. ACM. doi:10.1145/2608628.2608664.

15 Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, Cambridge, 2000.

16 Arnold Schönhage. Partial and total matrix multiplication. *SIAM Journal on Computing*, 10(3):434–455, 1981. doi:10.1137/0210032.

17 Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4):354–356, 1969. doi:10.1007/BF02165411.

18 Volker Strassen. Rank and optimal computation of generic tensors. *Linear algebra and its applications*, 52:645–685, 1983. doi:10.1016/0024-3795(83)80041-X.

19 Marcos Villagra, Masaki Nakanishi, Shigeru Yamashita, and Yasuhiko Nakashima. Tensor rank and strong quantum nondeterminism in multiparty communication. In *Theory and applications of models of computation*, volume 7287 of *Lecture Notes in Comput. Sci.*, pages 400–411. Springer, Heidelberg, 2012. arXiv:1202.6444, doi:10.1007/978-3-642-29952-0_39.

20 Péter Vrana and Matthias Christandl. Entanglement distillation from Greenberger-Horne-Zeilinger shares. *arXiv preprint arXiv:1603.03964*, 2016. arXiv:1603.03964.

21 Ronald de Wolf. Nondeterministic quantum query and communication complexities. *SIAM Journal on Computing*, 32(3):681–699, 2003. arXiv:cs/0001014, doi:10.1137/S0097539702407345.

Quantum Codes from High-Dimensional Manifolds

Matthew B. Hastings

Station Q, Microsoft Research, Santa Barbara and Quantum Architectures and Computation Group, Microsoft Research, Redmond, USA
mahastin@microsoft.com

Abstract

We construct toric codes on various high-dimensional manifolds. Assuming a conjecture in geometry we find families of quantum CSS stabilizer codes on N qubits with logarithmic weight stabilizers and distance $N^{1-\epsilon}$ for any $\epsilon > 0$. The conjecture is that there is a constant $C > 0$ such that for any n-dimensional torus $\mathbb{T}^n = \mathbb{R}^n/\Lambda$, where Λ is a lattice, the least volume unoriented $n/2$-dimensional cycle (using the Euclidean metric) representing nontrivial homology has volume at least C^n times the volume of the least volume $n/2$-dimensional hyperplane representing nontrivial homology; in fact, it would suffice to have this result for Λ an integral lattice with the cycle restricted to faces of a cubulation by unit hypercubes. The main technical result is an estimate of Rankin invariants[24] for certain random lattices, showing that in a certain sense they are optimal. Additionally, we construct codes with square-root distance, logarithmic weight stabilizers, and inverse polylogarithmic soundness factor (considered as quantum locally testable codes[1]). We also provide an short, alternative proof that the shortest vector in the exterior power of a lattice may be non-split[8].

1998 ACM Subject Classification E.4 Coding and Information Theory

Keywords and phrases quantum codes, random lattices, Rankin invariants

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.25

1 Introduction

Quantum CSS stabilizer codes[6] can be understood in terms of homology[18, 13, 4], and different manifolds provide a rich source of different codes. The two-dimensional toric code[18, 13] and four-dimensional toric code[9] are commonly considered examples; they are code families based on families of cellulations of a two and four dimensional tori. Other manifolds[14] provide other interesting properties, such as greater distance, discussed below. In this paper, we consider families of codes based on high dimensional manifolds.

We begin by considering some parameters that quantify a CSS code. The elementary degrees of freedom of a CSS codes are qubits (or, more generally, qudits, for some $d \geq 2$). Let there be N such qudits so that the Hilbert space has dimension d^N. CSS codes can be parametrized by several parameters, which we write as $[[N, K, D, W]]$. Here N is the number of qudits. K is the number of encoded qudits, so that the code has a code space which is a subspace of dimension d^K. D is the "distance" of the code, defined below, while W is the "weight" of the stabilizers, defined also below. Generally speaking, larger K and D is desirable, while smaller W is also desirable (this discussion of desirability of certain values of the parameters ignores other questions like the ability to efficiently decode or encode states, which is a completely separate discussion that we do not consider in this paper).

The best families of quantum codes obtained thus far have significantly worse scaling than the corresponding scaling for classical linear codes. Families of classical codes exist with $K = \Theta(N), D = \Theta(N), W = \mathcal{O}(1)$ (so-called low density parity check codes provide


© Matthew B. Hastings;
licensed under Creative Commons License CC-BY

such an example[15, 20]). If we set $W = \mathcal{O}(1)$, then the largest known distance for a quantum code family is $\Theta(\sqrt{N\log(N)})$as in Ref. [14], while if we want $D = \Theta(N)$, then the lowest known weight is $W = \Theta(\sqrt{N})$ as in Ref. [5]. These parameters refer to stabilizer codes; if one allows subsystem codes[23], then it is possible to achieve $D = \Theta(N^{1-\epsilon}), W = \mathcal{O}(1)$ for $\epsilon = \mathcal{O}(1/\sqrt{\log(N)})$ as in Ref. [3], but now the parameter W does not refer to the weight of a set of commuting stabilizers but rather the to weight of a set of generators of the "gauge group" and these generators need not commute with each other. If one requires that the stabilizer group be generated by local commuting operators, then currently no advantage is known for a subsystem code. Another notable stabilizer code family achieves $K = \Theta(N), D = \Theta(\sqrt{N}), W = \mathcal{O}(1)$ and has efficient an efficient local decoding algorithm[26].

In this paper, we construct code families that, assuming a conjecture in geometry, have almost linear distance and logarithmic weight generators. We review various concepts before giving an overview of the paper.

1.1 Review of CSS Codes and Relation to Homology

The code subspace is the subspace of the d^N-dimensional Hilbert space which is in the $+1$ eigenspace of several "stabilizers". These stabilizers are of two types, called "X-type" and "Z-type". The Z operator on the d-dimensional Hilbert space of a single qudit is the operator

$$Z = \begin{pmatrix} 1 & & & \\ & \exp(\frac{2\pi i}{d}) & & \\ & & \exp(\frac{4\pi i}{d}) & \\ & & & \dots \end{pmatrix}, \tag{1}$$

while the X operator is the operator

$$X = \begin{pmatrix} 0 & 1 & & \\ & 0 & 1 & \\ & & 0 & 1 \\ & & & \dots \\ 1 & 0\dots & & \end{pmatrix}. \tag{2}$$

We write Z_i or X_i to indicate the operator Z or X acting on qudit i, tensored with the identity on all other qudits. Then, a Z-type stabilizer is the tensor product of Z operators on some qubits, possibly raised to integer powers. Such a Z-type stabilizer might be written, for example, $Z_1 Z_3^2$ to indicate that it is the tensor product of Z on qubit 1 with the square of Z on qubit 3. These exponents all can be taken in the range $1, 2, ..., d-1$; if an operator on a given qubit is raised to power 0, we simply do not write it when writing the Z stabilzer. The X-type stabilizers are similar, with Z replaced by X.

We encode the Z-type stabilizers in a matrix that we denote ∂_2. This matrix has N rows and has one column per Z-type stabilizer. The entries of the matrix are over the field $\mathbb{F}_d$. The entry in the i-th row and j-th column indicates which power of Z_i appears in the j-th stabilizer; thus, for example, for the stabilizer $Z_1 Z_3^2$, the first row in the corresponding column would have a 1 and the third row would have a 2 and all other rows would be zero. We encode the X-type stabilizers also in a matrix, denoted by ∂_1. This matrix has N columns and one row per X-type stabilizer, again with the entries over the field $\mathbb{F}_d$. The entry in the i-th row and j-th column indicates which power of X_j appears in the i-th stabilizer

A final requirement on CSS codes is that the stabilizers commute with each other. Any pair of Z-type stabilizers trivially commute, as do any pair of X-type stabilizers. The re-

quirement that the Z-type stabilizer commute with the X-type stabilizers can be simply expressed in terms of ∂_2, ∂_1 as

$$\partial_1 \partial_2 = 0. \tag{3}$$

This requirement is equivalent to saying that there is a chain complex

$$\mathcal{C}_2 \stackrel{\partial_2}{\to} \mathcal{C}_1 \stackrel{\partial_1}{\to} \mathcal{C}_0,$$

where $\mathcal{C}_2, \mathcal{C}_1, \mathcal{C}_0$ are vector spaces over $\mathbb{F}_d$, with basis elements in one-to-one correspondence with Z-type stabilizers, qudits, and X-type stabilizers, respectively. We have $\dim(C_1) = N$.

The number of encoded qudits K is given by the first Betti number, which is equal to $N - \dim(\mathcal{C}_2) - \dim(\mathcal{C}_0)$ assuming that all stabilizers are independent of each other (i.e., that the columns of ∂_2 are linearly independent, as are the rows of ∂_1).

The distance D is defined as follows. Let us say that an operator O is a Z-type logical operator if it is a tensor product of Z operators on qudits which commutes with all X-type stabilizers and which is not itself a product of Z-type stabilizers. In the language of homology, such an operator is a representative of a nontrivial first homology class; write

$$O = \prod_i Z_i^{a_i},$$

where the product ranges over all qudits and a_i are in $\mathbb{F}_d$. Define an N-component vector v with entries a_i, so that the requirement that O commutes with all X-type stabilizers is that $\partial_1 v = 0$, while the requirement that O not be a product of Z-type stabilizers is that v is not in the image of ∂_2. An X-type logical operator is defined similarly, with Z and X integerchanged everywhere in the definition. The weight of a Z-type (or X-type) logical operator O is defined to be the number of qudits i such that Z_i (or X_i)appears in O raised to a nonvanishing power mod d; we say that that Z_i or X_i is in the support of the logical operator. We define D_Z to be the minimum weight of a Z-type logical operator and D_X to be the minimum weight of an X-type logical operator and define the distance D by

$$D = \min(D_X, D_Z). \tag{4}$$

We define the weight W of a code to be the least integer W such that every row and every column of ∂_2 has at most W nonvanishing entries and also every row and every column of ∂_1 has at most W nonvanishing entries. Note that this means that not only does every stabilizer act on at most W different qudits, also every qudit is acted on by at most W different Z-type stabilizers and W different X-type stabilizers.

We define the weight of an operator which is a product of Z and X operators to be the number of qudits on which the operator acts nontrivially; for example, the operator $X_1 X_3$ has weight 2. Thus, every stabilizer has weight at most W.

A vector v in a vector space $\mathcal{C}_k$ is called a k-chain (or simply, a "chain"). If $\partial_k v = 0$, then v is called a k-cycle. The weight of a vector is defined to be the number of nonzero entries in the vector.

1.2 CSS Codes from Manifolds and Systolic Freedom

Conversely, just as one can define a chain complex from a CSS code, one can use a chain complex to define a CSS code. Given any chain complex over some field $\mathbb{F}_d$, one can define a qudit CSS code: choose any vector space in the chain complex to correspond to the qudits,

and then the vector spaces of one higher and one lower dimension correspond to the Z-type and X-type stabilizers. For example, given a triangulation (or cubulation or other discretization) of a four dimensional manifold one can define a chain complex

$$\mathcal{C}_4 \stackrel{\partial_4}{\to} \mathcal{C}_3 \stackrel{\partial_3}{\to} \mathcal{C}_2 \stackrel{\partial_2}{\to} \mathcal{C}_1 \stackrel{\partial_1}{\to} \mathcal{C}_0,$$

where the basis elements of $\mathcal{C}_k$ correspond to k-cells. Then, one can choose any integer q and let the qudits correspond to the q-cells and the Z-type stabilizers correspond to $(q+1)$-cells and the X-type stabilizers correspond to $(q-1)$-cells. The case $q = 2$ is the familiar four-dimensional toric code of Ref. [9], while the cases $q = 0, 4$ are classical repetition codes (Ising models) in the Z or X basis, respectively.

Defining CSS codes from manifolds has several nice advantages. For one, often the distance of the code can be translated into geometric properties of the manifold and (up to some technical details that we discuss below) it can be geometrically interpreted as the least possible volume of a q-dimensional cycle in a nontrivial homology class. Similarly, if the triangulation has a bounded local geometry, then this gives a bound on W.

Naively, it might seem that such constructions will not be able to obtain a better-than-square-root distance, i.e. $D = \Omega(\sqrt{N})$. We now give some intution for this naive belief, and give a more detailed discussion of the relation between volume and number of qudits in one particular example, as it will be useful later. Consider an n-dimensional torus constructed from a hypercube of length ℓ on each side for some integer ℓ by gluing the opposite faces together. Introduce coordinates $(x_1, \ldots, x_n)$. Discretize the torus by hypercubes of unit length in the obvious way, so that the 0-cells are at integer values of the coordinates. In this case, the volume of the torus equals the number of hypercubes in the discretization, which equals ℓ^n. The number of qudits is given by

$$N = \ell^n \binom{n}{q},$$

while

$$D_Z = \ell^q, \quad D_X = \ell^{n-q}.$$

To see that $D_Z \leq \ell^q$, one can pick any q-dimensional plane where q of the coordinates assume arbitrary values and the other coordinates are held fixed at integer values; then, the product of Z over the q-cells in this plane give a logical operator. We omit the proof that this upper bound for D is tight in this case. The value of D_X is given by picking any $(n-q)$-dimensional plane on the dual lattice and then taking the product of X over the the q-cells that intersects this plane also gives a logical operator.

Choosing the optimal value, $q = n/2$ still leads only to $D = \Theta(\sqrt{N})$. Varying the geometry of the torus by changing the aspect ratio (i.e., keeping the sides of the torus orthogonal to each other but changing the relative lengths) does not lead to any improvement.

However, this naive belief is false. "Systolic freedom" is the term for a concept due to Gromov[2], that one may have manifolds for which the product of the q-systole (the least volume cycle representing a nontrivial element of q-th homology) times the $(n-q)$-systole may be arbitrarily larger than the volume of the manifold. This phenomenon was originally observed for integer homology (corresponding to qudit quantum codes with large d), while only later in Ref. [14] was it constructed for $\mathbb{Z}_2$ homology.

1.3 Overview of Paper

In the original construction of systolic freedom[2], the topology of the manifold was held fixed and the metric was varied to obtain a diverging ratio between the product of the q-

systole and the $(n-q)$-systole to the volume of the manifold (for some given pairs $q, n-q$), while in the $\mathbb{Z}_2$ case[14], the topology of the manifold was varied to obtain a diverging ratio. In this paper, we consider instead a family of manifolds with different dimension. Most of the paper is devoted to considering tori $\mathbb{R}^n/\Lambda$ for certain random lattices Λ. In section 2 we make various definitions of the random lattices and define Rankin invariants. In section 3 we give an overview of the construction and present a geometric conjecture 1 and state theorem 5 that, assuming the conjecture, there exist quantum CSS codes with logarithmic weight and almost linear distance. In section 4 we prove lower bounds on the Rankin invariant of certain random lattices, which is the main step in proving theorem 5. In section 5 we discuss some obstacles to proving even a weaker form of conjecture 1 (involving cycles with integer coefficients) and we consider shortest vectors in the exterior product of a lattice. Finally, in section 6 we give some alternative constructions which have only square-root distance but which have inverse polylogarithmic soundness parameters as quantum locally testable codes[1].

To give some motivation to our lattice construction, consider the two-dimensional toric code. On a square lattice with length ℓ on each side, there are $2\ell^2$ qubits and the distance ℓ. Suppose we ignore the details of the cellulation and take an arbitrary torus $\mathbb{R}^2/\Lambda$, pretending that the number of qubits is equal to the area (ℓ^2) and the distance is equal to the shortest vector in the lattice Λ. Then, a slightly better geometry than the square lattice would be to take the hexagonal lattice, as the ratio of the square of the length of the shortest vector to the area of the torus is equal to $2/\sqrt{3}$ rather than 1. This is only a slight constant improvement over the square lattice. However, in higher dimensions, the shortest vector in lattice Λ can be roughly $\sqrt{n}$ longer than the $1/n$ power of the volume of the torus $\mathbb{R}^n/\Lambda$. Further, if we consider least volume cycles representing nontrivial homology for $q > 1$, then larger improvements are possible (at least for cycles which are hyperplanes). This motivates our construction and the consideration of so-called "Rankin invariants"[24].

In this paper, we will consider codes based on cellulations of such tori (in higher dimensions) where the sides of the torus are not orthogonal. These tori will be $\mathbb{R}^n/L$ for some lattice L, where the lattice does not have a basis of orthogonal vectors. However, L will be an integral lattice, so that it is still possible to cellulate $\mathbb{R}^n/L$ by unit hypercubes. This is done by taking the obvious cellulation of $\mathbb{R}^n$ by unit hypercubes (i.e., every unit hypercube whose vertices are at integer points is a cell in the cellulation) and then identifying cells that differ by translation by a lattice vector. Fig. 1 gives a simple example in the case $n = 2$.

2 Random Lattices and Definitions

Consider a so-called LDA lattice[7, 10] as follows. We pick a prime p. We will construct a lattice which is a subset of $\mathbb{Z}^n$ for some even n. We first construct a linear code over field $\mathbb{F}_p^n$. We define this code by a "code generator matrix" G which is an n-by-k matrix such that the *column* vectors are a basis for the codewords. (We explicitly call it a "code generator matrix" rather than just a "generator matrix", as we will also consider lattice generator matrices later.) Usually in coding theory, it is instead conventional to let the *rows* of a code generator matrix be the basis for a code, but to maintain consistency with notation we use later, we instead use the *columns* as the basis. We choose the entries of G independently and uniformly from $\mathbb{F}_p$.

We will be interested in taking n large at fixed ratio $k/n < 1$. With high probability (i.e., with probability tending to 1 as $n \to \infty$ with k/n fixed), G is non-degenerate (see next paragraph). Assuming that G is indeed non-degenerate, one can find a permutation of the

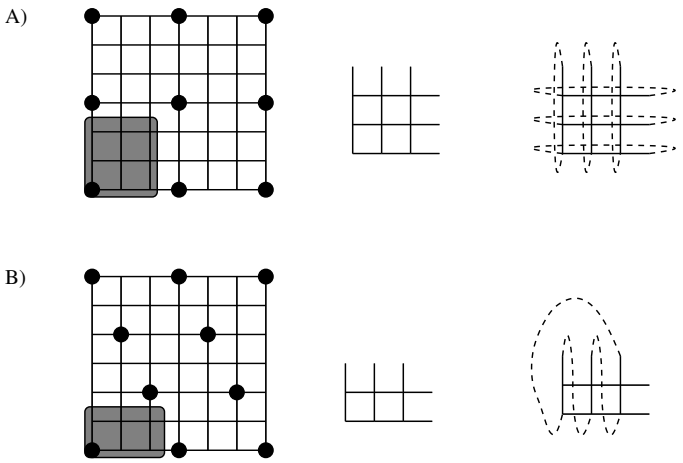


Figure 1 Leftmost part of A) shows part of a cellulation of $\mathbb{R}^2$ by unit squares, as well as showing a lattice L with basis vectors $(3,0)$ and $(0,3)$. Solid circles represent points of L. The shaded region contains a maximal set of points in $\mathbb{Z}^2$ which do not differ by translation by L. Middle part of A) shows 0-cells of the shaded region, as well as 1- and 2-cells attached to them. Rightmost part of A) shows how the 1-cells in the middle part are attached: the dashed lines indicate that the top of a given 1-cell on the top of the figure is attached to a 0-cell at the bottom. Similarly for attaching cells on the right to those on the left. Figure B) is similar, but now L has basis vectors $(3,0)$ and $(2,2)$. Now the dashed lines indicate that a 1-cell on the top of the figure is attached to a 0-cell on the bottom, but that 0-cell is translated horizontally (horizontal dashed lines are not shown in B; they are the same as in A). For both A,B, the code has distance 3, but B has smaller volume.

rows such that G is in the form

$$G = \begin{pmatrix} A \\ B \end{pmatrix}$$

where A, B are k-by-k and $(n-k)$-by-k matrices with A non-degenerate. Then, since A is non-degenerate there exists a sequence of elementary column operations that brings A to the identity matrix, where for a matrix over $\mathbb{F}_p^n$ an elementary column operation is one of: adding one column to another, multiplying a column by any nonzero element of the field, or interchanging two columns. These column operations bring G to the form

$$G = \begin{pmatrix} I \\ C \end{pmatrix},$$

where I is the k-by-k identity matrix and C is some $(n-k)$-by-k matrix. Since the entries of C are obtained by applying these column operations to the entries of B, the entries of C are chosen independently of each other and uniformly from $\mathbb{F}_p$, i.e., applying any elementary column operation to an ensemble of matrices with entries chosen uniformly and independently leaves this ensemble invariant. This is the form of G that we work with in the rest of this section.

▶ **Definition 1.** Let the lattice L_0 be the set of points $x_1, ..., x_n$ in $\mathbb{Z}^n$ such that the vector $(x_1 \mod p, ..., x_n \mod p)$ is in the linear code defined by G.

We now show that with high probability, G is non-degenerate. With probability $1 - (1/p)^n$, the first column of G has a nonzero entry. By elementary column operations, adding a multiple of the first column to other columns, we can set all other columns equal to zero

in the first row for which the first column has a nonzero entry. Then, with probability $1-(1/p)^{n-1}$, the second column has a nonzero entry in some other row. Add a multiple of the second column to the third, fourth,... column to set them equal to zero in the first row for which the second column has a nonzero entry. Continuing in this fashion, the probability that G is non-degenerate is $(1-(1/p)^n)(1-(1/p)^{n-1})\dots(1-(1/p)^{n-k+1})$ which indeed is $1-o(1)$.

The lattice L_0 is the set of integer linear combinations of the columns of G (interpreted as vectors of integers, rather than as vectors of elements of $\mathbb{F}_p$) and of the n vectors with a p in one coordinate and zeroes elsewhere. Then, the lattice L_0 is the set of integer linear combinations of the columns of the matrix

$$\begin{pmatrix} I & pI & 0 \\ C & 0 & pI \end{pmatrix},$$

where the row blocks have sizes k and $n-k$ respectively, while the column blocks have sizes k, k, $n-k$, and respectively, and where I is the identity matrix of appropriate size. However, any integer linear combination of column vectors of $\begin{pmatrix} pI \\ 0 \end{pmatrix}$ is also an integer linear combination of column vectors of

$$\begin{pmatrix} I & 0 \\ C & pI \end{pmatrix}.$$

To see this, consider any vector of integers $\vec{y} = (y_1, ..., y_k)$. Then,

$$\begin{pmatrix} pI \\ 0 \end{pmatrix} \vec{y} \quad = \quad \begin{pmatrix} I \\ C \end{pmatrix} p\vec{y} - \begin{pmatrix} 0 \\ pI \end{pmatrix} C\vec{y}. \tag{5}$$

Thus, L_0 is the set of integer linear combinations of columns of the matrix

$$B_0 = \begin{pmatrix} I & 0 \\ C & pI \end{pmatrix}.$$

A matrix B such that the lattice is the set of integer combinations of columns of B is called a generating matrix for the lattice. Two different generating matrices B_1, B_2 define the same lattice if and only if $B_1 = B_2 T$ where T is an integer matrix such that T^{-1} also is an integer matrix. In this case, the matrix B_1 can be turned into the matrix B_2 by a sequence of elementary column operations where an elementary column operations is one of: adding one column to another, changing the signs of all entries in a column, or interchanging two columns.

Given a lattice L with generating matrix B which is an n-by-k matrix, such that B has rank k, we define the volume of the lattice to equal $\mathrm{vol}(L) = \det(B^\dagger B)^{1/2}$. If $k = n$, then $\mathrm{vol}(L) = |\det(B)|$.

▶ **Definition 2.** Given any linearly independent set of vectors $x_1, ..., x_k$ in $\mathbb{Z}^n$ (or more generally in $\mathbb{R}^n$) we define their volume $\mathrm{vol}(x_1, ..., x_k)$ to be the volume of the lattice generated by the n-by-k matrix with columns $x_1, ..., x_k$.

This matrix B_0 is lower triangular and so $\det(B_0)$ is easily computed:

$$\mathrm{vol}(L_0) = |\det(B_0)| = p^{n-k}. \tag{6}$$

▶ **Definition 3.** An "integral lattice" is defined to be a lattice whose generating matrix has integer entries. A "primitive lattice" is defined to be an integral lattice such that there is no

other integral lattice of the same rank properly containing it, where the rank of a lattice is defined to be the dimension of the subspace spanned by points in the lattice. Equivalently, a primitive lattice is an integral lattice such that there is no integral lattice which spans the same subspace and properly contains it.

Example: in two dimensions, the lattice generated by the vector $(2, 1)$ is primitive, while that generated by $(4, 2)$ is not.

Unless specified, all lattices will be in n dimensions. We use $|\dots|$ to denote the ℓ_2 norm of a vector.

Finally, we define the Rankin invariant.

▶ **Definition 4.** The Rankin invariant $\gamma_{n,m}(L)$ for a lattice L with rank n is defined to be

$$\gamma_{n,m}(L) = \min_{\substack{v_1,\dots,v_m \in L \\ \mathrm{vol}(v_1,\dots,v_m)\neq 0}} \left(\frac{\mathrm{vol}(v_1,\dots,v_m)}{\mathrm{vol}(L)^{m/n}}\right)^2. \tag{7}$$

The square in the above definition is included for historical reasons. The factor m/n in the exponent of $\mathrm{vol}(L)$ is such that the invariant is unchanged under rescaling the lattice L by any constant factor. In the case $m = 1$, the Rankin invariants $\gamma_{n,1}(L)$ is related to the length of the shortest vector: $\gamma_{n,1}(L) = \min_{x\in L, x\neq 0} \frac{|x|^2}{\mathrm{vol}(L)^{2/n}}$. Clearly, $\gamma_{n,n}(L) = 1$ for all L.

The Rankin invariant $\gamma_{n,1}(L)$ is related to the length of the shortest vector in the lattice. To understand the higher Rankin invariants, consider a set of vectors $v_1, ..., v_m \in L$ with $\mathrm{vol}(v_1, ..., v_m) \neq 0$. Consider the torus $\mathbb{R}^n/L$. The m-dimensional hyperplane spanned by $v_1, ..., v_m$ represents a nontrivial integer homology class and has an m-dimensional volume (using the Euclidean metric) equal to $\mathrm{vol}(v_1, ..., v_m)$.

One can also understand the higher Rankin invariants in another way. A choice of vectors $v_1, ..., v_m \in L$ with $\mathrm{vol}(v_1, ..., v_m) \neq 0$ corresponds to a basis for a rank-m sublattice of L. Thus, the Rankin invariants are a lower bound on the volume of rank-m sublattices. This interpretation will be used later, in section 4.

3 Overview of Construction: Conjectures and Main Result on Distance

We will consider a family of CSS codes obtained by choosing a fixed $p > 1$ and taking LDA lattices with $k = n/2$ from the random ensemble above, for all (even) values of n. With high probability, this lattice has rank n. Given the integral lattice L_0, we take a cellulation of the torus $\mathbb{R}^n/L_0$ by hypercubes of length 1 on each side. Then, we consider a qubit toric code on this cellulation with degrees of freedom on q-cells for $q = n/2$. Then, the number of q-cells is equal to

$$N = \binom{n}{n/2} p^{n/2}. \tag{8}$$

The distance of the code is equal to the weight of the least weight logical X or Z operator. The vector corresponding to such an operator represents nontrivial homology or cohomology with $\mathbb{Z}_2$ coefficients. We conjecture that:

▶ **Conjecture 1.** *There exists a constant $C > 0$, such that for any n-dimensional integral lattice L, for the toric code obtained by the cellulation using integer hypercubes and degrees of freedom on q-cells for $q = n/2$, the distance is lower bounded by*

$$C^n \min_{\substack{v_1,\dots,v_q \in L \\ \mathrm{vol}(v_1,\dots,v_q)\neq 0}} \mathrm{vol}(v_1, ..., v_q) = C^n \mathrm{vol}(L)^{q/n} \gamma_{n,q}(L)^{1/2}$$

.

Let us motivate this conjecture. The least volume hyperplane representing nontrivial homology has volume equal to the Rankin invariant. This hyperplane need not lie on the q-cells that we have chosen. We can deform the hyperplane to get a cycle that lies on the q-cells using the Federer-Fleming deformation theorem[11]: this theorem is based on deforming the cycle to lie on the $(n-1)$-skeleton (i.e., the $(n-1)$-dimensional faces of the hypercubes of unit size), then on the $(n-2)$-skeleton, and so on, iteratively, until the cycle lies on the q-skeleton. The deformation to move cycle from the m-skeleton to the $(m-1)$-skeleton is done by choosing a point randomly in an m-dimensional hypercube and then projecting the cycle outwards from that point to the boundary. This deformation may increase the volume, but that is fine: what we are considered with is lower bounding the volume.

However, it is not clear that the optimal operator is obtained by such a deformation procedure starting from a hyperplane. There may be, for example, unoriented chains which are not hyperplanes but which represent nontrivial homology and have much smaller volume than the least volume hyperplane. The conjecture is that such cycles can have at most exponentially smaller (i.e., smaller by a factor C^n) volume.

Conjecture 1 considers the distance of the code, which is equal to the least volume of a $\mathbb{Z}_2$ cycle representing nontrivial homology. The cycles are in the chain complex obtained from the cellulation using hypercubes. One may be tempted to make a (possibly stronger) conjecture that a similar inequality holds for more general chains, such as smooth chains. This conjecture has a purely geometrical statement:

▶ **Conjecture 2.** *There exists a constant $C > 0$, such that for any n-dimensional lattice L, the $n/2$-systole with $\mathbb{Z}_2$ coefficients of the torus $\mathbb{R}^n/L$ is lower bounded by*

$$C^n \min_{\substack{v_1,\ldots,v_q \in L \\ \mathrm{vol}(v_1,\ldots,v_q)\neq 0}} \mathrm{vol}(v_1,...,v_q) = C^n \mathrm{vol}(L)^{q/n} \gamma_{n,q}(L)^{1/2} \ .$$

Conjecture 1 would follow from conjecture 2 since the systole is defined by taking the infimum over smooth cycles, while in conjecture 1 we restrict to cycles which lie on the cellulation by unit hypercubes. In this regard, we remark that the possible increase in volume from the Federer-Fleming deformation theorem may be superexponentially large: the upper bound is at most $2n^{n/2}\binom{n}{n/2}$ (see Ref. [12]).

In this paper, we prove that:

▶ **Theorem 5.** *Assume that conjecture 1 holds. Then, for any $\epsilon > 0$, there exists a family of quantum CSS codes on N qubits with distance $D = \Omega(N^{1-\epsilon})$ and weight $w = \mathcal{O}(\log(N))$ and with $\Theta(N^\delta)$ encoded qubits, where $\delta > 0$ (δ depends on ϵ).*

This theorem will follow from a corollary of theorem 23, which implies that for any constant $c < 1/\sqrt{2\pi e}$, with high probability we have $\min_{\substack{v_1,\ldots,v_q \in L \\ \mathrm{vol}(v_1,\ldots,v_q)\neq 0}} \mathrm{vol}(v_1,...,v_q) \geq (cp)^{n/2}$. Hence, with high probability, $D \geq (cC^2p)^{n/2}$. Since $N = \binom{n}{n/2}p^{n/2} \leq (4p)^{n/2}$, with high probability we have

$$D \geq (cC^2p)^{\log_{4p}(N)} = N^{\frac{\log(cC^2p)}{\log(4p)}} .$$

Fixing c to be any constant slightly smaller than $1/\sqrt{2\pi e}$, we find that for any $\epsilon > 0$ that for all sufficiently large p we have

$$1 - \epsilon \leq \frac{\log(cC^2p)}{\log(4p)}$$

so that $D \geq N^{1-\epsilon}$.

We have $w = \mathcal{O}(d) = \mathcal{O}(\log(N))$.

The number of encoded qubits is equal to $\binom{n}{k} = 2^{(1-o(1))n} = 2^{2(1-(o(1))\log_{4p}(N)} = N^{2(1-o(1))/\log(4p)} \equiv N^{\delta}$.

The main work will be theorem 23, to lower bound the Rankin invariant for this class of lattices. The reader may wonder why we introduce this class of lattices, instead of re-using previous results which show that there exist random lattices with a large Rankin invariant, $\gamma_{n,n/2}(L) \geq (\frac{k}{12})^{n/4}$. See theorem 3 in Ref. [16]. The reason is that the random lattices constructed there need not be integral lattices and so we do not have such an obvious cell decomposition to place on the lattices. We comment later on the relationship between the Rankin invariant for our lattice (which depends on n, p) and the invariant in Ref. [16]; this requires considering how large n needs to be compared to p in our construction.

Note that we choose p large so that the exponentially growing factor, $\approx 2^n$, arising from the factor $\binom{n}{n/2}$ in the number of cells will be polynomially smaller than the volume $p^{n/2}$. We have $2^n = (p^n)^{1/\log_2(p)}$.

We remark that similar code constructions can be made by choosing degrees of freedom on q-cells for $q \neq n/2$, taking n large at a fixed ratio q/n. In this case, a natural generalization of conjecture 1 is to assume that there is a constant C such that $d_Z \geq C^n \mathrm{vol}(L)^{q/n} \gamma_{n,q}(L)^{1/2}$ and $d_X \geq C^n \mathrm{vol}(L)^{(n-q)/n} \gamma_{n,n-q}(L)^{1/2}$. Assuming this conjecture, our construction would give a code with $d_X d_Z$ polynomially larger than N.

4 Rankin Invariants

In this section, we will prove lower bounds on the Rankin invariants[24] $\gamma_{n,m}(L_0)$ of L_0. The proof uses the probabilistic method; in particular, we use the first moment method. To motivate the proof, let us first sketch a proof method for $\gamma_{n,1}(L_0)$; then, we give a sketch a possible extension of the proof method to $\gamma_{n,m}(L_0)$ and explain some difficulties with this extension; finally, we outline the approach we use which is a modification of that. First, suppose we just want to lower bound $\gamma_{n,1}(L_0)$; i.e., we wish to lower bound the shortest vector in the lattice. This can be done by a first moment method: estimate the number of integer vectors with length less than some given length ℓ; then, compute the probability that any given integer vector is in the lattice (this probability is $p^{-(n-k)}$ for a randomly chosen code assuming G is non-degenerate); so, for sufficiently small ℓ, the average number of integer vectors with length less than ℓ in the lattice is small so it is unlikely that any integer vectors with length less than ℓ will be in the lattice. One might attempt to do something similar for the Rankin invariants: estimate the number of rank m integral lattices in n dimensions with volume at most V and then compute the probability that an integral lattice is in a randomly chosen linear code, i.e., that this integral lattice is a sublattice of L_0. Call this rank-m lattice K and call its generating matrix M_K. In fact, Ref. [25] provides asymptotic estimates (large V) for the number of such lattices K, so it might seem that one could directly use the results there in a first moment method. Indeed, this approach might work, but since the results of Ref. [25] hold in the asymptotic limit (large V), some additional estimates would be needed (we do use many results in Ref. [25]). However, the results we need are in some ways simpler than that of Ref. [25] because we do not care about an exact estimate of the number of such lattices, only an upper bound. Further, rather than applying the first moment method by estimating the number of lattices K with some given volume and estimating the probability that such a lattice is in the code and then showing that the product is small for small V, we will apply the first moment method to each column of the generating matrix M_K *separately* (with M_K written in Hermite normal form). That is, we

first estimate (this step is exactly analogous to the discussion at the start of this paragraph regarding how to lower bound $\gamma_{n,1}(L_0)$) the probability that there is a choice for the first column which has small length and which is in the code. Then, we estimate the probability that there is a choice for the second column which is also in the code such that the ratio of the volume of the lattice generated by the first two columns of M_K to the volume of the lattice generated by the first column of M_K is small. To do this calculation, we need the concept of "factor lattice" [25], reviewed below. We continue in this fashion over the other columns, showing that the ratio of the volume of the lattice generated by the first a rows of M_K to the volume of the lattice generated by the first $a-1$ rows of a is likely to be large, for each $a = 2, 3, \ldots$.

4.1 Counting Points

Let $V_d(r)$ denote the volume of a ball of radius r in d dimensions:

$$V_d(r) = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)} r^d. \tag{9}$$

Given a rank-l lattice L which spans some space E, we define the Voronoi cell to be the set of points y in E such that $|y| \leq |y-v|$ for all lattice points $v \neq 0$. The l-dimensional volume of the Voronoi cell is equal to $\mathrm{vol}(L)$.

▶ **Definition 6.** Given a lattice L, let $N(L, z, r)$ denote the number of points in lattice L within distance r of some given point z.

▶ **Lemma 7.** *Let L be a rank-l lattice in d dimensions which spans some space E. Suppose the diameter of the Voronoi cell of L is bounded by some given D. Then, for any z, r,*

$$N(L, z, r) \leq \frac{1}{\mathrm{vol}(L)} V_l(r+D). \tag{10}$$

Proof. For every point $x \in L$ within distance r of z, let T_x be the set of points $y \in E$ such that $y-x$ is in the interior of the Voronoi cell. The sets T_x are non-overlapping and each has l-dimensional volume $\mathrm{vol}(L)$. So, the volume of $\cup_{x,|x-z|\leq r} T_x$ is equal to $N(L, z, r)\mathrm{vol}(L)$. Every x is within distance r of z and so every point in $\cup_{x,|x-z|\leq r} T_x$ is within distance $r+D$ of z, so $N(L, z, r)\mathrm{vol}(L) \leq V_k(r+D)$. ◀

We make some more definitions.

▶ **Definition 8.** Given a rank-l lattice L spanning a subspace E, the polar lattice L^P is the lattice of all vectors in E which have integral inner products with all vectors in L. The polar lattice also has rank l and $\mathrm{vol}(L^P)\mathrm{vol}(L) = 1$.

▶ **Definition 9.** Let Γ_0^n be the rank-n lattice in n dimensions consisting of all vectors for which all coordinates are integral.

▶ **Definition 10.** Given a primitive lattice L spanning subspace E, the orthogonal lattice $L^\perp$ consists of all vectors in Γ_0^n with vanishing inner product with all vectors in L.

▶ **Definition 11.** Let L be a rank-l primitive sublattice of Γ_0^n and let E be the subspace spanned by L. Let π project onto the orthogonal complement of E, which we write $E^\perp$. Let $\pi(\Gamma_0^n) \equiv \Gamma_0^n / L$. Then, Γ_0^n / L is also a lattice, called the factor lattice. It has rank $n-l$

We have[25]

$$\mathrm{vol}(L)\mathrm{vol}(\Gamma_0^n/L) = 1. \tag{11}$$

This equation follows from this lemma:

▶ **Lemma 12.**

$$\Gamma_0^n/L = ((L)^\perp)^P. \tag{12}$$

Proof. See Ref. [25]. ◀

▶ **Lemma 13.** *Let L be a rank-l primitive sublattice of Γ_0^n. Let π and Γ_0^n/L be as above. Then, the diameter of the Voronoi cell of Γ_0^n/L is bounded by $\sqrt{n-l}$.*

Proof. Since L has rank $l < n$, there must be some vector w_1 which has a 1 in one coordinate and zeroes in all other coordinates (i.e., w_1 is of the form $(0, \ldots, 0, 1, 0, \ldots, 0)$) which is not in E. Then, since the span of E and w_1 has dimension $l+1$, if $k < n-1$, there must be some other vector w_2 of the same form which is not in the span of E and l_1. Proceeding in this fashion, we construct vectors $w_1, ..., w_{n-l}$, all of which have zeroes in all but one coordinate and a 1 in that coordinate. The vectors $\pi(w_i)$ span $E^\perp$. So, every point y in $E^\perp$ can be written as a linear combination $y = \pi(\sum_i a_i w_i)$. If the a_i are integer, then y is a lattice point in $\pi(\Gamma_0^n)$. Every linear combination $\sum_i a_i w_i$ is within distance $(1/2)\sqrt{n-l}$ of some linear combination $\sum_i b_i w_i$ with integer b_i (to see this, simply round all a_i to the nearest integer). Since the norm does not increase under projection, every $\pi(\sum_i a_i w_i)$ is also within distance $(1/2)\sqrt{n-l}$ of some $\pi(\sum_i b_i w_i)$ for integer b_i and hence every point in E is within distance $(1/2)\sqrt{n-l}$ of a lattice point. ◀

We remark that the lattice with basis vectors $\pi(w_i)$ may not include all points in $\pi(\Gamma_0^n)$; as an example, consider $l = 1$ and $n = 2$ and let L be the lattice with basis vector $(2, 1)$ and let $w_1 = (0, 1)$. The vector $\pi((1, 0))$ is then not included in the lattice with basis vector $\pi(w_1)$.

▶ **Lemma 14.** *Let L be a rank-l primitive sublattice of Γ_0^n. Let π and Γ_0^n/L be as above. The number of points in Γ_0^n/L within distance r of the origin is bounded by*

$$N(\Gamma_0^n/L, 0, r) \leq \mathrm{vol}(L)V_l(r + \sqrt{n-l}). \tag{13}$$

Proof. This follows from lemmas 7,13 and Eq. (11). ◀

4.2 Hermite Normal Form For Lattices

Consider a rank-m integral lattice K in n dimensions. If this lattice has basis vectors $v_1, ..., v_m$, we write an n-by-m matrix M_K whose columns are these basis vectors. We label the rows of the matrix by integers $1, \ldots, n$ and label the columns by integers $1, \ldots, m$. Such a matrix is called a lattice generator matrix for the lattice. Then, the set of points in the integral lattice is the image under M_K of Γ_0^m. By a sequence of column operations (adding one column of M_K to another column, which does not change the image, or changing the sign of a column, which also does not change the image), we can bring always bring the matrix M_K to so-called "Hermite normal form"; further, there is a unique matrix M_K in Hermite normal form which generates K.

Our definition of Hermite normal form differs from that of other authors because we will *reverse* the order of columns and *reverse* the order of rows compared to the usual order. This is because we will be doing induction later and with the reversed order of columns, the notation will be much more natural later. See Eq. (17) for an example of Hermite normal form below.

▶ **Definition 15.** A matrix M_K is said to be in Hermite normal form if for every column j there is a row i_j with $1 \leq i_1 < i_2 < \ldots < i_m \leq n$ such that the entries of M_K obey:

$$i > i_j \quad \rightarrow \quad (M_K)_{i,j} = 0 \tag{14}$$

and

$$(M_K)_{i_j,j} > 0, \tag{15}$$

and

$$l > j \quad \rightarrow \quad 0 \leq (M_K)_{i_j,l} < (M_K)_{i_j,j}. \tag{16}$$

We say that "the first a columns of M_K are in Hermite normal form" if the submatrix of M_K consisting of the first a columns is in Hermite normal form. In this case, for every column j with $j \leq a$ there is a row i_j with $1 \leq i_1 < i_2 < \ldots < i_a \leq n$ such that Eqs. (14,15,16) hold when restricted to the case that $j \leq a$ and $l \leq a$.

We introduce some notation. This notation defines various vector spaces and vectors in terms of the matrix M_K; we do not explicitly write M_K in the definition, but rather the particular choice of M_K should be clear in context. The last nonzero entry in the j-th column occurs in the i_j-th row. Define a sequence of lattices $K_1, K_2, ..., K_m$, where K_j has rank j and K_j is defined to be the lattice generated by the submatrix of M_K containing the first j rows and the first i_j columns. Note that $K_m = K$. Note also that if K_a is primitive then K_b is primitive for all $b < a$.

We let $\vec{v}_j$ be the vector given by the first i_j rows of the j-th column.

This notation can be clarified with an example of $n = 5, m = 3$, with $i_1 = 2, i_2 = 4, i_3 = 5$, where we write

$$M_K = \begin{pmatrix} (\vec{v}_1)_1 & (\vec{v}_2)_1 & (\vec{v}_3)_1 \\ (\vec{v}_1)_2 & (\vec{v}_2)_2 & (\vec{v}_3)_2 \\ 0 & (\vec{v}_2)_3 & (\vec{v}_3)_3 \\ 0 & (\vec{v}_2)_4 & (\vec{v}_3)_4 \\ 0 & 0 & (\vec{v}_3)_5 \end{pmatrix}, \tag{17}$$

with $(\vec{v}_j)_i$ denoting the i-th entry of vector $\vec{v}_j$. For this matrix to be in Hermite normal form, we have $0 \leq (\vec{v}_2)_2, (\vec{v}_3)_2 < (\vec{v}_1)_2$ and $0 \leq (\vec{v}_3)_4 < (\vec{v}_2)_4$.

The lattice K_j is a sublattice of $\Gamma_0^{i_j}$. We let M_{K_j} be the submatrix of M_K consisting of the first j rows and the first i_j columns so that M_{K_j} generates K_j. We also define a lattice $\tilde{K}_j$ which is a sublattice of $\Gamma_0^{i_{j+1}}$. The lattice $\tilde{K}_j$ will be the sublattice generated by the submatrix of M_K consisting of the first j rows and the first i_{j+1} columns. We let $M_{\tilde{K}_j}$ be the submatrix of M_K consisting of the first j rows and the first i_{j+1} columns. Hence, the last $i_{j+1} - i_j$ entries of every vector in $\tilde{K}_j$ are equal to 0.

Let π_j project onto the orthogonal complement of the span of $\tilde{K}_j$.

▶ **Lemma 16.** *Let K be a rank-m lattice in n dimensions with generating matrix M_K in Hermite normal form. Then, there exist an n-by-m integer matrix M_{K^P} which is a lattice generating matrix in Hermite normal form (with the same i_j as M_K) for a primitive lattice, and an m-by-m integer matrix F which is upper triangular with positive diagonal entries such that we have*

$$M_K = M_{K^P} F. \tag{18}$$

Further, F, M_{K^P} are unique.

Proof. Let K^P be a primitive lattice spanning the same space as K and containing the lattice K. (Note that such a primitive K^P must exist and is unique: it is the lattice consisting of all integer points which are in the space spanned by K). Let K^P be generated by M_{K^P} with M_{K^P} in Hermite normal form; note that since K^P is unique, M_{K^P} is uniquely determined by K. Then, since K is contained in K^P, every column of M_K is an integer linear combination of columns of M_{K^P}. So, $M_K = M_{K^P}F$ for some integer matrix F.

M_K, M_{K^P} must have the same i_j or their columns would not span the same space.

Since M_K, M_{K^P} have the same i_j, it follows that F is upper triangular with positive diagonal entries: restrict M_K, M_{K^P} to the rows $i_1, i_2, \ldots, i_m$, giving upper triangular matrices of size m-by-m. Call these matrices A, B respectively. Then $A = BF$, so $F = B^{-1}A$. Since B is upper triangular, so is B^{-1} and so is F. ◀

4.3 Counting Column Choices

▶ **Definition 17.** A lattice L is *consistent* with a code generator matrix G if every point $(x_1, ..., x_n)$ in the lattice has the property that $(x_1 \mod p, \ldots, x_n \mod p)$ is in the code defined by G. A lattice generator matrix M_L is consistent with a code generator matrix G if the lattice generated by M_L is consistent with G.

We will use a to label a column choice, $1 \leq a \leq m$. We will construct lattices K_a in terms of K_{a-1} and $\vec{v}_a$.

▶ **Lemma 18.** *Let M_K be in Hermite normal form. Then,*

$$\mathrm{vol}(K_a) = \mathrm{vol}(K_{a-1})|\pi_{a-1}(\vec{v}_a)|. \tag{19}$$

Proof. Immediate from the definition of volume. ◀

Assume K_{a-1} is primitve. Then, the next lemma gives a one-to-one correspondence between vectors $\vec{v}_a$ obeying *one* of the conditions needed for Hermite normal form (the condition Eq. (16)) and vectors in a certain factor lattice. In lemma 20 we consider the case that K_{a-1} is not primitive. Note that there is an additional condition on v_a, namely that its first entry be positive, in order for the matrix M_{K_a} to be in Hermite normal form.

▶ **Lemma 19.** *Let the first $a-1$ columns of M_K be given and assume that the first $a-1$ columns of M_K are in Hermite normal form and assume that K_{a-1} is a primitive sublattice of Γ_0^n. Then, there is a one-to-one correspondence between vectors $\vec{v}_a$ such that*

$$j < a \quad \rightarrow \quad 0 \leq (M_K)_{i_j,a} < (M_K)_{i_j,j} \tag{20}$$

and points $\vec{x}$ of the lattice $\Gamma_0^{i_a}/\tilde{K}_{a-1}$, such that if $\vec{x}$ corresponds to $\vec{v}_a$ then $\pi_{a-1}(\vec{v}_a) = \vec{x}$.

Proof. We will show that for every $\vec{x} \in \Gamma_0^{i_a}/\tilde{K}_{a-1}$, there exists a unique $\vec{v}_a$ obeying Eq. (20) such that $\pi_{a-1}(\vec{v}_a) = \vec{x}$. This gives a map $\mathcal{F}$ from $\Gamma_0^{i_a}/\tilde{K}_{a-1}$ to vectors obeying Eq. (20). This map is one-to-one since distinct vectors $\vec{x}_1 \neq \vec{x}_2$ cannot both be the image of the same vector $\vec{v}_a$ under the map π_{a-1}. This map $\mathcal{F}$ is onto since any vector $\vec{v}_a$ obeying Eq. (20) is the image of $\pi_{a-1}(\vec{v}_a)$ under this map.

First we show existence of somer vector $\vec{v}_a$. Every vector $\vec{x}$ in $\Gamma_0^{i_a}/\tilde{K}_{a-1}$ is given by $\vec{x} = \pi_{a-1}(\vec{y})$ for some $\vec{y} \in \Gamma_0^{i_a}$. For any such vector $\vec{y}$, we can add lattice vectors in $\tilde{K}_{a-1}$ so that Eq. (20) (i.e., set $\vec{v}_a$ equal to $\vec{y}$ plus some sum of lattice vectors; this can be done iteratively, so that it holds first for $j = a-1$, then $j = a-2$, and so on). Adding these lattice vectors does not change the image of the result under π_{a-1}.

Now uniqueness. Suppose that $\pi_{a-1}(\vec{y}) = \pi_{a-1}(\vec{z})$ for $\vec{y}, \vec{z}$ being two possible choices of $\vec{v}_a$ such that Eq. (20) is obeyed. Then, $\pi_{a-1}(\vec{y}-\vec{z}) = 0$, so $\vec{y}-\vec{z}$ is in the span of $\tilde{K}_{a-1}$. Since K_{a-1} is primitive so is $\tilde{K}_{a-1}$ and so $\vec{y}-\vec{z}$ is in $\tilde{K}_{a-1}$. Let $M_K(i,j)$ denote the submatrix of M_K containing the first i rows and the first j columns, so that $M_K(i_a, a-1)$ is a lattice generating matrix for $\tilde{K}_{a-1}$. So, $\vec{y}-\vec{z} = M_K(i_a, a-1)\vec{u}$, where $\vec{u} \in \Gamma_0^{a-1}$. Then, Eq. (20) requires that $\vec{u} = 0$. This follows inductively: if the last entry of $\vec{u}$ is nonzero, then it is not possible for both $\vec{y}$ and $\vec{z}$ to obey Eq. (20) for $j = a-1$; to see this, note that then the $(a-1)$-th entries of $\vec{y}, \vec{z}$ must differ by a positive integer multiple of $(M_K)_{i_j,j}$ and so they cannot both fall in the range $0, 1, \ldots, (M_K)_{i_j,j} - 1$. So, $\vec{y}-\vec{z}$ differs by an elements of the lattice generated by $M_K(i_a, a-2)$ and so $\vec{y}-\vec{z} = M_K(i_a, a-2)\vec{u}'$ for $\vec{u}' \in \Gamma_0^{a-2}$. Again, the last entry of $\vec{u}'$ must equal zero so that Eq. (20) will be obeyed for $j = a-2$. We continue inductively for $j = a-3, \ldots$. ◀

The next lemma is similar to the previous except that we no longer assume that K_{a-1} is primitive.

▶ **Lemma 20.** *Let the first $a-1$ columns of M_K be given and assume that the first $a-1$ columns of M_K are in Hermite normal form.*

Let $M_K(i,j)$ denote the submatrix of M_K containing the first i rows and the first j columns, so that $M_{\tilde{K}_{a-1}} = M_K(i_a, a-1)$ is a lattice generating matrix for $\tilde{K}_{a-1}$. Use lemma 16 to write

$$M_{\tilde{K}_{a-1}} = M_{\tilde{K}^P_{a-1}} F.$$

Then, the possible choices of $\vec{v}_a$ such that

$$j < a \quad \rightarrow \quad 0 \leq (M_K)_{i_j,a} < (M_K)_{i_j,j} \tag{21}$$

are in one-to-one correspondence with choices of tuples $(\vec{x}, f_1, \ldots, f_{a-1})$, where $\vec{x}$ is a point in $\Gamma_0^{i_a}/\tilde{K}^P_{a-1}$ and $f_1, \ldots, f_{a-1}$ are integers obeying $0 \leq f_i < F_{i,i}$, such that if $(\vec{x}, f_1, \ldots, f_{a-1})$ corresponds to $\vec{v}_a$ then $\pi_{a-1}(\vec{v}_a) = \vec{x}$. Thus, there are $\det(F)$ distinct vectors $\vec{v}_a$ corresponding to $\vec{x}$.

Proof. We will show that for every $\vec{x} \in \Gamma_0^{i_a}/\tilde{K}_{a-1}$, there exist $\det(F)$ distinct vectors $\vec{v}_a$ obeying Eq. (21) such that $\pi_{a-1}(\vec{v}_a) = \vec{x}$. These $\det(F)$ vectors will be labelled by $f_1, \ldots, f_{a-1}$.

First we show existence. Every vector $\vec{x}$ in $\Gamma_0^{i_a}/\tilde{K}_{a-1}$ is given by $\vec{x} = \pi_{a-1}(\vec{y})$ for some $\vec{y} \in \Gamma_0^{i_a}$. For any such vector $\vec{y}$, we can add lattice vectors in $\tilde{K}_{a-1}$ so that Eq. (21) will hold (this can be done iteratively, so that it holds first for $j = a-1$, then $j = a-2$, and so on). Adding these lattice vectors does not change the image of the result under π_{a-1}.

Now, for each $\vec{x}$, let $\vec{z}$ be some fixed vector such that Eq. (21) holds for $\vec{v}_a = \vec{z}$ and such that $\vec{x} = \pi_{a-1}(\vec{z})$. Suppose that $\pi_{a-1}(\vec{y}) = \pi_{a-1}(\vec{z})$ for $\vec{y}$ some other possible choice of $\vec{v}_a$ such that Eq. (21) is obeyed. We count the number of possible choices of $\vec{y}$. Then, $\pi_{a-1}(\vec{y}-\vec{z}) = 0$, so $\vec{y}-\vec{z}$ is in the span of $\tilde{K}_{a-1}$. Since $\tilde{K}^P_{a-1}$ is primitive, $\vec{y}-\vec{z} = M_{\tilde{K}^P_{a-1}}\vec{u}$, where $\vec{u} \in \Gamma_0^{a-1}$. There are $F_{a-1,a-1}$ possible choices for the $(a-1)$-th entry of $\vec{u}$. To see this, note that $\vec{y}$ and $\vec{z}$ both obey Eq. (21) for $j = a-1$. For $j = a-1$, this equation gives a constraint that the $(a-1)$-th entry of $\vec{y}$ must fall in the range $0, \ldots, (M_K)_{i_{a-1},a-1} - 1$. The $(a-1)$-th entry of $\vec{y}$ is determined by the $(a-1)$-th entry of $\vec{u}$ and shifting that entry of $\vec{u}$ by one shifts the $(a-1)$-th entry of $\vec{y}$ by $(M_{\tilde{K}^P_{a-1}})_{i_{a-1},a-1}$. We have $(M_K)_{i_{a-1},a-1} = (M_{\tilde{K}^P_{a-1}})_{i_{a-1},a-1}F_{a-1,a-1}$ so that there are $F_{a-1,a-1}$ possible choices. Then, given this choice of the $(a-1)$-th entry of $\vec{u}$, there are $F_{a-2,a-2}$ possible choices for the $(a-2)$-th entry of $\vec{u}$, and so on. ◀

▶ **Lemma 21.** *Let M_K be a matrix in Hermite normal form which is a lattice generating matrix for a rank-m integral lattice K in n dimensions. Let K_{a-1} be given and let r be a real number. Let $C(r, K_{a-1})$ denote the number of choices of K_a such that*

$$\mathrm{vol}(K_a) \leq rK_{a-1}. \tag{22}$$

Then,

$$C(r, K_{a-1}) \leq \mathrm{vol}(K_{a-1})V_{i_a-a+1}(r + \sqrt{i_a - a + 1}). \tag{23}$$

If $r < 1$ then $C(r, K_{a-1}) = 0$.

Proof. Let $\vec{v}_a$ be as defined above. By lemma 18

$$\mathrm{vol}(K_a) = \mathrm{vol}(K_{a-1})|\pi_{a-1}(\vec{v}_a)|. \tag{24}$$

so $|\pi_{a-1}(v_a)| \leq r$. By Eq. (15), the first entry of $\vec{v}_a$ is ≥ 1, and since all vectors in K_{a-1} vanish in the first entry, we have $|\pi_{a-1}(\vec{v}_a)| \geq 1$, so indeed $C(r, K_{k-1}) = 0$ for $r < 1$.

By lemma 20, the vector $\vec{v}_a$ is in one-to-one correspondence with a tuple $(\vec{x}, f_1, \ldots, f_{a-1})$ where $\vec{x}$ is a vector in lattice $\Gamma_0^{i_a}/\tilde{K}_{a-1}^P$. By lemma 13, the lattice $\Gamma_0^{i_a}/\tilde{K}_{a-1}^P$ has the diameter of its Voronoi cells bounded by $\sqrt{i_a - a + 1}$. So, for given $\Gamma_0^{i_a}/\tilde{K}_{a-1}^P$ and given r, the number of possible choices of $\vec{x}$ such $|\pi_{a-1}(\vec{v}_a)| \leq r$ is bounded by

$$N(\Gamma_0^{i_a}/\tilde{K}_{a-1}^P, 0, r) \leq \frac{1}{\mathrm{vol}(\Gamma_0^{i_a}/\tilde{K}_{a-1}^P)} V_{i_a-a+1}(r + \sqrt{i_a - a + 1}). \tag{25}$$

So, by Eq. (11),

$$N(\Gamma_0^{i_a}/\tilde{K}_{a-1}^P, 0, r) \leq \mathrm{vol}(K_{a-1}^P)V_{i_a-a+1}(r + \sqrt{i_a - a + 1}). \tag{26}$$

Factorize $M_K(i_a, a-1) = M_{\tilde{K}_{a-1}^P} F$, as in lemma 20.

The number of possible choices of $f_1, \ldots, f_{a-1}$ is equal to $\det(F) = \mathrm{vol}(K_{a-1})/\mathrm{vol}(K_{a-1}^P)$.

So, the total number of choices of K_a is bounded by

$$\det(F)\mathrm{vol}(K_{a-1}^P)V_{i_a-a+1}(r + \sqrt{i_a - a + 1}) = \mathrm{vol}(K_{a-1})V_{i_a-a+1}(r + \sqrt{i_a - a + 1}), \tag{27}$$

as claimed. ◀

4.4 First Moment Bound

▶ **Lemma 22.** *Let G be an n-by-k code generator matrix for a code, chosen from the ensemble defined previously (entries chosen independently and uniformly from $\mathbb{F}_p$). Let M_K be an n-by-k lattice generator matrix. Let K_{a-1} be given and assume the first $a-1$ columns of M_K are in Hermite normal form. Let $Pr(K_{a-1}, r)$ denote the probability that, conditioned on K_{a-1} being consistent with G, there exists a choice of v_a such that K_a is consistent with G and such that the first a columns of M_K are in Hermite normal form and such that*

$$\mathrm{vol}(K_a) \leq rK_{a-1}. \tag{28}$$

Then, for $r < 1$, $Pr(K_{a-1}, r) = 0$, and for $r < p$,

$$Pr(K_{a-1}, r) \leq p^{-(n-k)}\mathrm{vol}(K_{a-1})V_{i_a-a+1}(r + \sqrt{i_a - a + 1}). \tag{29}$$

Proof. By lemma 21, indeed there are no choices of v_a such that $r < 1$. If $r < p$, then $0 < (\vec{v}_a)_1 < p$ so $(\vec{v}_a)_1 \neq 0 \mod p$. So, the a-th column of M_K is not in the span of the first $a-1$ columns of M_K modulo p. So, even though we have conditioned on K_{a-1} being consistent with G, the probability that a given choice of $\vec{v}_a$ is consistent with G is bounded by $p^{-(n-k)}$. (The probability is $p^{-(n-k)}$ if we condition on G being non-degenerate and smaller if G may be degenerate.)

So, by lemma 21, the average number of choices of $\vec{v}_k$ consistent with G is bounded by $p^{-(n-k)}\mathrm{vol}(K_{a-1})V_{i_a-a+1}(r+\sqrt{i_a-a+1})$. ◀

The next theorem estimates the probability that, for a randomly chosen code generator matrix, there is a rank-m lattice K of small volume which is consistent with that matrix. The bounds becomes effective for volume smaller than $(cp)^{\min(m,n-k)}$ with $c < 1/\sqrt{2\pi e}$.

▶ **Theorem 23.** *Let $P_{lat}(H,p,n,m)$ denote the probability that for a random code generator matrix G for a code over $\mathbb{F}_p^n$ there is a rank-m lattice K consistent with a code generator matrix such that* $\mathrm{vol}(K) \leq H$.

For any p, for any real number $x > \sqrt{2\pi e}$, for sufficiently large $n-m$,

$$P_{lat}((cp)^{\min(m,n-k)},p,n,m) \leq mc^m x^{n-m+1}. \tag{30}$$

The required $n-m$ is quadratic in $p(x-\sqrt{2\pi e})^{-1}$.

Proof. Note that if there is a lattice K of rank-m consistent with the code generator matrix, then the lattices $K_1,\ldots,K_{m-1}$ constructed above have ranks $1,\ldots,m-1$ respectively and are also consistent with the code generator matrix and have $\mathrm{vol}(K_a) \leq \mathrm{vol}(K)$. So, it suffices to consider the case $m \leq n-k$ (if $m > n-k$, then consider the lattice K_{n-k} instead).

For M_K in Hermite normal form, since $i_1 < i_2 < \ldots < i_m < n$, we have $i_a \leq n-m+a$ and so $i_a - a + 1 \leq n-m+1$. We use the bound (the inequality on the second line holds for all sufficiently large $n-m$)

$$\begin{aligned} V_{n-m+1}(r+\sqrt{n-m+1}) &= \frac{\pi^{\frac{n-m+1}{2}}}{\Gamma(\frac{n-m+1}{2})}(r+\sqrt{n-m+1})^{n-m+1} \\ &\leq \left(\frac{2\pi e}{n-m+1}\right)^{\frac{n-m+1}{2}}(r+\sqrt{n-m+1})^{n-m+1} \\ &= \left(\frac{r\sqrt{2\pi e}}{\sqrt{n-m+1}}+\sqrt{2\pi e}\right)^{n-m+1}, \end{aligned} \tag{31}$$

Let c be a real number, $0 < c < 1$. We will make a choice of c below.

By lemma 22 and Eq. (31), given K_{a-1}, if $\mathrm{vol}(K_{a-1}) \leq (cp)^m$, we have

$$\begin{aligned} Pr(K_{a-1},r) &\leq c^m p^{m-(n-k)}\left(\frac{r\sqrt{2\pi e}}{\sqrt{n-m+1}}+\sqrt{2\pi e}\right)^{n-m+1} \\ &\leq c^m\left(\frac{r\sqrt{2\pi e}}{\sqrt{n-m+1}}+\sqrt{2\pi e}\right)^{n-m+1}. \end{aligned} \tag{32}$$

For $r < p$, this is bounded by $c^m\left(\frac{p\sqrt{2\pi e}}{\sqrt{n-m+1}}+\sqrt{2\pi e}\right)^{n-m+1}$. For any p, for any real number $x > \sqrt{2\pi e}$, for sufficiently large $n-m$, this is bounded by $c^m x^{n-m+1}$. (The required $n-m$ is quadratic in $p(x-\sqrt{2\pi e})^{-1}$).

Suppose that $\mathrm{vol}(K) \leq (cp)^m$ for some $c < 1$. Then, $\mathrm{vol}(K_a) \leq (cp)^m$ for all a and for some a we have $\mathrm{vol}(K_a)/\mathrm{vol}(K_{a-1}) < p$. However, for $\mathrm{vol}(K_a) \leq (cp)^m$, the above calculation bounds the probability for given a that there is a choice of K_a such that

$\mathrm{vol}(K_a)/\mathrm{vol}(K_{a-1}) < p$ by $c^m x^{n-m+1}$ for all sufficiently large $n-m$. By a union bound, the probability that for some a there is a choice of K_a such that $\mathrm{vol}(K_a)/\mathrm{vol}(K_{a-1}) < p$ is bounded by $mc^m x^{n-m+1}$ for all sufficiently large $n-m$. So, $P_{lat}((cp)^m, p, n, m) \leq mc^m x^{n-m+1}$ for all sufficiently large $n-m$. ◀

This implies the following corollary for the Rankin invariant:

▶ **Corollary 24.** *For any p, k, for all sufficiently large n at fixed ratio m/n, for any $c < 1/\sqrt{2\pi e}$, with high probability we have*

$$\gamma_{n,m}(L_0) \geq (cp)^{2\min(m,n-k)} p^{-2m(n-k)/n}. \tag{33}$$

(Recall that with high probability G is non-degenerate so L_0 is rank n.)

We remark that it might be possible to tighten the bounds of theorem 23 to bound P_{lat} even for some range of x smaller than $\sqrt{2\pi e}$, especially for small m. One possible way to tighten the bounds is to use the fact that if there $\mathrm{vol}(K) < p^{m-z}$ for some integer $z > 0$ then there must be at least z different a such that $\mathrm{vol}(K_a)/\mathrm{vol}(K_{a-1}) < p$; in the proof above we only used that there was at least one such a.

We remark also that, up to the constant c, the value of the Rankin invariant at $m = k = n/2$ is optimal for an integral lattice; i.e., the dependence on p is optimal. The reason is that it implies that an $n/2$-dimensional sublattice of L_0 has the same volume (again, up to factors of c^m) as L_0 does.

It is also worth comparing the value of the Rankin invariant that we find to the Rankin invariant for random lattices (from a different ensemble) in Ref. [16]. The Rankin constant $\gamma_{n,m}$ is defined to be the maximum of $\gamma_{n,m}(L)$ over all lattices L. Those random lattices in Ref. [16] were used to lower bound the Rankin constant $\gamma_{n,n/2}$ by $\gamma_{n,n/2} \geq (\frac{k}{12})^{n/4}$. Since we need to take $n \sim p^2$ for the bounds of theorem 23 to be effective, if we choose $m = k = n/2$ and $p \sim \sqrt{n}$ we find that with high probability $\gamma_{n,n/2}(L_0) \geq (\text{const.} \times n)^{n/4}$. Thus, we find the same leading behavior $n^{n/4}$, with the Rankin invariants differing only by factors const.^{n}.

5 Volume of Oriented Systole

In this section, we consider a weaker conjecture than 1. Throughout this section, we consider the case of homology using integer coefficients, rather than $\mathbb{Z}_2$ coefficients. In this setting, there is a general method, called "calibration" [17] for lower bounding weights. We will show that this method gives an effective lower bound for homology classes which have a particular form, which we call "split", but we will show that it does not give a useful lower bound in general. The reason for this is related to the existence of short vectors in the exterior q-th power of L_0.

Given an rank-n lattice L, we write its m-th exterior power as $\wedge^m L$. This exterior power is a lattice of vectors in $\binom{n}{m}$ dimensions; the vectors in this lattice are linear combinations (with integer coefficients) of vectors $v_1 \wedge v_2 \wedge \ldots \wedge v_m$, where $v_i \in L$ and the exterior product is anti-symmetric under interchange: $v_1 \wedge v_2 = -v_2 \wedge v_1$.

▶ **Definition 25.** A vector v in $\wedge^m L$ is called "split" if $v = x_1 \wedge \ldots \wedge x_m$ for $x_1, \ldots x_m \in L$.

The q-th homology classes of the torus T^n are in one-to-one correspondence with vectors in $\wedge^q \mathbb{Z}^n$. For the torus $\mathbb{R}^n/L_0$ that we consider, it will be more convenient to regard the classes as being in one-to-one correspondence with vectors in $\wedge^q L_0$. That is, the k-th homology class represented by a hyperplane which is a span of k basis vectors will correspond to the vector which is the exterior product of these k basis vectors.

The lattice $\wedge^m L$ inherits an inner product:

$$(x_1 \wedge \ldots \wedge x_m) \cdot (y_1 \wedge \ldots \wedge y_m) = \det(S),$$

where S has matrix elements $S_{i,j} = x_i \cdot y_j$. We write this norm $|X|$, where $X \in \wedge^m L_0$. Calibration allows one to lower bound the volume of a representative of a homology class in $\wedge^q L_0$ using this inner product.

We first explain this lower bound in the split case. The arguments are not new.

▶ **Lemma 26.** *Let* $\mathrm{vol}(v_1, \ldots, v_q) \neq 0$. *Then, the minimum volume of any closed chain (either a sum of faces of q-faces of the unit hypercubes used in the cubulation or more generally an arbitrary sum of simplices) representing homology class* $X = v_1 \wedge \ldots \wedge v_q$ *is greater than or equal to* $|v_1 \wedge \ldots \wedge v_q|$.

Proof. Let us write $v \cdot d\vec{x}$ to denote a differential 1-form $\sum_i (v)_i dx^i$, where $i = 1, \ldots, n$ are orthogonal basis directions in Euclidean space and $(v)_i$ are components of i. Consider the differerential q-form $\omega = (v_1 \cdot d\vec{x}) \wedge (v_2 \cdot d\vec{x}) \wedge \ldots \wedge (v_q \cdot d\vec{x})$. Let S denote the hyperplane spanned by vectors $v_1, \ldots, v_q$ (the hyperplane is oriented, so the order of vectors matters). We have $\int_S \omega = |X|^2$. Further, for any chain C in the same homology class as S, we have $\int_C \omega = \int_S \omega = |X|^2$, where the integral over C is given by writing C as a sum of q-faces of the unit hypercubes and integrating ω over each face. (Indeed, one can also consider more general C, such as sums of arbitrary simplices, and the same result holds). For a q-face (or indeed any sum of q-dimensional simplices), the integral of ω over that face is bounded by $|X|$ times the volume of the face. Hence, the volume of C must be at least equal to $(\int_C \omega)/|X| = |X|$. ◀

Now we consider the nonsplit case. In contrast to the split case where we were able to "calibrate" the hyperplane S (find a differential form assuming maximum value on that hyperplane), we might not be able to calibrate nonsplit homology classes. However, we can still obtain a lower bound.

▶ **Lemma 27.** *Let* $X \in \wedge^q L$, $X \neq 0$. *Then, the minimum volume of any closed chain representing homology class X is lower bounded by* $|X|$.

Proof. Write $X = \sum_a X_a$, where X_a are split vectors. For each $X_a = v_1^a \wedge \ldots \wedge v_q^a$, define a differential q-form $\omega_a = (v_1^a \cdot d\vec{x}) \wedge \ldots \wedge (v_q^a \cdot d\vec{x})$. Let $\omega = \sum_a \omega_a$.

Let S_a denote the hyperplane spanned by vectors $v_1^a, \ldots, v_q^a$. Let S denote the union of hyperplanes S_a. We have $\int_{S_a} \omega_b = (X_a, X_b)$. Hence, $\int_S \omega = |X|^2$.

We now consider the maximum of the integral of C over a q-face or q-dimensional simplex of unit volume. This is equal to

$$\max_{V \text{ split}, |V|=1} (V, X),$$

where we take the maximum over all split vectors $V \in \wedge^q \mathbb{R}^n$, with V not necessarily in $\wedge^q L$; i.e., $V = v_1 \wedge \ldots \wedge v_q$ for *arbitrary* $v_1, \ldots v_q$, with $v_1, \ldots, v_q$ not necessarily in the lattice L (i.e., we are upper bounding the integral over a unit volume square in the hyperplane spanned by $v_1, \ldots v_q$). If we relax the requirement that V be split, we have $\max_V (V, X) = |X|$. The restriction to split V can only reduce the maximum, so the maximum over split V is at most $|X|$. So, as in lemma 26, since the integral over ω over any chain representing the same homology class as X must be equal to the $\int_S \omega = |X|$, the volume of such a chain must be at least $|X|$. ◀

One may wonder whether the bound in lemma 27 can be significantly improved if we do not relax the requirement that V be split. Of course, if X is split, then

$$\max_{V \text{ split}, |V|=1}(V, X) \geq |X|/\sqrt{\binom{n}{q}} = |X|$$

and the maximum is achieved for $V = X$. However, for X not split, the maximum might be smaller and so the lower bound on the volume would be correspondingly: we can lower bound the volume of a closed chain representing homology class X by $|X|^2/\max_{V \text{ split}, |V|=1}(V, X)$. Unfortunately, this at best only leads to a small improvement in the bound. We claim that

$$\max_{V \text{ split}, |V|=1}(V, X) \geq |X|/\sqrt{\binom{n}{q}}, \tag{34}$$

so that at best we would lower bound the volume by $\sqrt{\binom{n}{q}}|X|$, and since $\sqrt{\binom{n}{q}} < 2^{n/2}$, this leads to only a small improvement (recall that there are $N = p^{n/2}$ qubits and we choose $p >> 1$). To see Eq. 34, consider the orthogonal basis for $\wedge^q\mathbb{R}^n$ of vectors $x_1 \wedge \ldots \wedge x_q$ where $x_1, \ldots, x_q$ are chosen from the n different coordinate directions. These basis vectors are all split. Since $\wedge^q\mathbb{R}^n$ is $\binom{n}{q}$-dimensional, there must be some basis vector V such that $|(V, X)| \geq |X|/\sqrt{\binom{n}{q}}$. Using this vector V (or its negation if the inner product (V, X) is negative) in the maximum gives Eq. (34).

The Rankin invariant is the minimal value of the norm $|X|$ over nonzero split vectors. Thus, the results on the Rankin invariant give a lower bound on the volume of representatives of split homology classes. However, in Ref. [8], it was shown that for certain lattices L the shortest nonzero vector in $\wedge^m L$ may be shorter than the Rankin invariant. Interestingly, the lattices we consider here provide another example where this occurs; in fact this occurs for any lattice with sufficiently large Rankin invariant.

▶ **Lemma 28.** *Let L be a rank-n lattice. Then, the shortest nonzero vector in $\wedge^m L$ has norm at most $\sqrt{\gamma_{\binom{n}{m}}}\text{vol}(L)^{m/n}$, where $\gamma_{\binom{n}{m}}$ denotes Hermite's constant in dimension $\binom{n}{m}$.*

Hence, if $\gamma_{n,m}(L) \geq \gamma_{\binom{n}{m}}$, then the shortest vector is not split.

Proof. We have $\text{vol}(\wedge^m L) = \text{vol}(L)^{\binom{n-1}{m-1}}$ by Proposition 1.10.4 of Ref. [21]. The lattice $\wedge^m L$ has rank $r = \binom{n}{m}$, and so the shortest nonzero vector in $\wedge^m L$ has length at most $\sqrt{\gamma_r}\text{vol}(\wedge^m L)^{1/r}$, where γ_r is Hermite's constant. So, the shortest nonzero vector in $\wedge^m L$ has length at most

$$\sqrt{\gamma_r}\text{vol}(L)^{\binom{n-1}{m-1}/\binom{n}{m}} = \sqrt{\gamma_r}\text{vol}(L)^{m/n}. \tag{35}$$

◀

For all r, we have $\gamma_r \leq 1 + r/4$, with an asymptotic behavior $\gamma_r \lesssim \frac{2r}{\pi e}$[22]. So, $\sqrt{\gamma_{\binom{n}{m}}} \leq \sqrt{1 + \binom{n}{m}/4}$. So, lemma 28 has an interesting interpretation for the application to quantum codes. If the bound in lemma 27 is saturated so that the least volume cycle representing a homology class has volume $|X|$, then we find that the code has roughly square-root distance. Thus, conjecture 1 implies that for some homology classes, the bound of lemma 27 is far from saturated. The possible improvement of Eq. (34) leads to only a small improvement here (though, it is possible that if the possible improvement of Eq. (34) holds for the homology classes with smallest $|X|$ and if the bound of lemma 28 is saturated then one might be able to prove a slightly above square-root distance for integer homology).

6 Quantum Locally Testable Codes from High-Dimensional Constructions

In this section, we give a construction of quantum codes which are "locally testable" [1] using high-dimensional constructions. The construction uses a different topology than above; the similarity in the constructions is simply that in both cases we consider a family of codes derived from manifolds of varying dimension, with the number of qubits in the code depending exponentially on the dimension of the manifold.

Let us write $\mathrm{wt}(O)$ to indicate the weight of an operator O. Similarly, given a vector v (in one of the vector spaces defining the chain complexes), we let $\mathrm{wt}(v)$ denote the number of nonzero entries in v.

Given a CSS stabilizer code defined from a chain complex $\ldots \mathcal{C}_{q+1} \stackrel{\partial_{q+1}}{\rightarrow} \mathcal{C}_q \stackrel{\partial_q}{\rightarrow} \mathcal{C}_{q-1} \ldots$, with the qudits associated with q-cells and the Z-type and X-type stabilizers associated with $(q+1)$-cells and $(q-1)$-cells, respectively, we define soundness parameters $\epsilon_X(w), \epsilon_Z(w)$ as follows:

▶ **Definition 29.** Define

$$\epsilon_Z(w) = \min_{v \in \mathcal{C}_q, \mathrm{wt}(v)=w, \partial_q v \neq 0} \left(\max_{u \in \mathcal{C}_q, \partial_q u = 0} \frac{\mathrm{wt}(\partial v)}{\mathrm{wt}(v+u)} \right). \tag{36}$$

Define $\epsilon_X(w)$ similarly, with ∂_q replaced with ∂_{q+1}^T, where the superscript T denotes transpose.

Equivalently, consider the minimum over all Z-type operators O, such that O has weight w and such that O does not commute with at least one stabilizer, of the following quantity: take the maximum, over all Z-type operators P which commute with all stabilizers, of the ratio of the number of stabilizers which do not commute with O to the weight of $O+P$. This minimum is ϵ_Z.

The codes of Ref. [26] have distance $\Theta(\sqrt{N})$, stabilizer weight $\mathcal{O}(1)$ and have $\epsilon_{X,Z}(w)$ bounded away from zero for $w \lesssim \sqrt{N}$, as shown in Ref. [19]. It is unclear whether or not families of codes exist which have distance which is $\Omega(\sqrt{N})$ and stabilizer weight $\mathcal{O}(1)$ and which have $\epsilon_{X,Z}(w)$ bounded away from zero for *all* w.

Here we give a simple construction of a family of qubit codes with 2 encoded qubits and with distance $\Theta(\sqrt{N})$, $\epsilon_{X,Z}(w)$ only polylogarithmically small for all w, and with *logarithmic* weight stabilizers. We warm up with a construction of a qubit code family with no encoded qubits (and hence the notion of distance is meaningless for this code) but with $\epsilon_{X,Z}(w)$ bounded away from zero for all w and with logarithmic weight stabilizers; we call this the "simplex code". We then give the full construction, which is based on a product of hyperspheres.

6.1 Simplex Code

Of course, with no encoded qubits, there are some fairly trivial constructions of code with $\epsilon_{X,Z}$ strictly bounded away from zero. For example, one can take a code with N qubits and stabilizers $Z_1, Z_2, \ldots, Z_N$. Thus, every product of Z operators commutes with all stabilizers (and so $\epsilon_Z(w)$ is a minimum over an empty set), while clearly $\epsilon_X(w) = 1$ for all w. However, the simplex code construction that we give obeys Poincare duality and has an entangled ground state.

The code we consider is obtained by taking a toric code on a n-dimensional sphere, with the degrees of freedom on q-cells for $q = n/2$. The exact value of q is not very important;

the important thing is that q/n is neither close to 0 nor close to 1 so that the number of q-cells will be exponential in n. However, the case $q = n/2$ is the self-dual case so this makes the proofs slightly simpler as we need to consider only one type of stabilizer.

The cellulation of the n-sphere that we use is to take the boundary of a $n+1$-dimensional simplex. We label the 0-cells by integers $1, \ldots, n+2$. For $0 \leq r \leq n$, there are $\binom{n+2}{r+1}$ distinct r-cells, labelled by subsets of $\Lambda \equiv \{1, \ldots, n+2\}$ with $r+1$ elements. We use qubits so the vector spaces are all over $\mathbb{F}_2$.

For $1 \leq r \leq n$, the boundary operator ∂_r acting on an r-cell labelled by some $(r+1)$-element set $S \subset \Lambda$ gives the sum of $r+1$ different $(r-1)$-cells, labelled by the distinct r-element subsets of S. For example, for $n \geq 2$, $\partial_2\{1,2,3\} = \{1,2\} + \{1,3\} + \{2,3\}$. We set $\partial_0 = 0$. One may verify that $\partial_{r-1}\partial_r = 0$ for all r.

For $q = n/2$, there are $N = \binom{n+2}{n/2+1}$ qubits, so N is exponentially large in n. Remark: in previous sections, the number of qubits we also had an exponential factor p^{n-k} which, for large p, was the dominant exponential scaling; in this subsection, we do not have such a factor.

Each qubit is acted on by $q+1$ stabilizers (as each q-cell has $q+1$ cells in its boundary) and each stabilizer acts on $q+2$ different qubits (as each $(q+1)$-cell has $q+2$ cells in its boundary and each $(q-1)$-cell has $q+2$ cells in its coboundary). Hence, the weight is indeed logarithmic in N, $w = (1/2 + o(1)) \cdot \log_2(N)$.

Finally, we show soundness. First, let us introduce notation.

▶ **Definition 30.** Given an r-cell σ labelled by some set S and a set $T \subset \Lambda$, we define $r \cup T$ to equal 0 if $S \cap T \neq \emptyset$ and otherwise $r \cup T$ is the $r + |T|$-cell labelled by $S \cup T$.

Given a vector $v \in \mathcal{C}_r$, we define $v \cup T$ by linearity. $v \cup T \in \mathcal{C}_{r+|T|}$ and the coefficient of $v \cup S$ corresponding to an $r + |T|$-cell labelled by a set U is equal to the coefficient of v corresponding to the r-cell labelled by $U \setminus T$ if $T \subset U$ and is equal to 0 is $T \not\subset U$.

▶ **Lemma 31.** *For the simplex code, for all w, $\epsilon_X(w) = \epsilon_Z(w) \geq 1$.*

Proof. Consider any $v \in \mathcal{C}_q$ with $\partial_q v \neq 0$. Set $w = (\partial_q v) \cup \{1\}$. Then, one may verify that $\partial_q x = \partial_q v$ (and hence, setting $w = x - v$, $\partial_q w = 0$) and that $\mathrm{wt}(x) \leq \mathrm{wt}(\partial_q v)$. ◀

The proof of soundness above has a very simple geometric interpretation. We take the boundary $\partial_q v$ and shrink it to a point (arbitrarily choosing the vertex $\{1\}$ as the point that we shrink it to).

6.2 Hypersphere Product Code

The above construction had constant soundness, but had no encoded qubits. We now give a different construction with 2 encoded qubits and distance $\sqrt{N}$ and inverse polylogarithmic soundness. We now consider the toric code on a product of spheres, $S^n \times S^n$.

We pick an integer $p \geq 1$ (p need not be prime); p will be chosen to equal $\log(N)$ below in order to achieve square-root distance. We choose a cellulation of S^n as follows: consider an $(n+1)$-dimensional hypercube of side length p on each side (we call this the "large" hyercube). Cellulate that large hypercube using hypercubes of side length 1 on each side; we call these the "small" hypercubes) (so that there are p^{n+1} small hypercubes in the cellulation). Then, take the boundary of the hypercube to get a cellulation of S^n.

A small $(n+1)$-dimensional hypercube has $\binom{n+1}{r}2^{n+1-r}$ different r-cells in its boundary (each r-cell is a product of 1-cells in r out of the $n+1$ directions and then for each of the remaining directions there are 2 possible choices of 0-cells). The number of r-cells in the

cellulation of the large hypercube is $\binom{n+1}{r}(p+1)^{n+1-r}p^r$. To see this, assign coordinates $[0,p]$ for each side of the large hypercube. Then, each r-cell is a product of 1-cells in r out of the $n+1$ directions with 0-cells in the remaining directions. The midpoints of the 1-cells are at half-integer coordinate in the interval $[0,p]$ and so there are p possible choices for each cell. There are $p+1$ possible choices for the coordinates of each 0-cell as these cells are at integer coordinates in the interval $[0,p]$. To determine the number of r-cells in the boundary of the large hypercube, restrict to the case that in at least one of the directions, the coordinate must be 0 or p. This gives the number equal to

$$\binom{n+1}{r}(p+1)^{n+1-r}p^r\Big(1-(\frac{p+1-2}{p+1})^{n+1-r}\Big),$$

where the ratio in parenthesis $\frac{p+1-2}{p+1}$ is the probability that for a random integer coordinate in the range $[0,p]$, the coordinate is not on the boundary 0 or p. Thus, there are at most $2^{(1-o(1))\cdot n}p^n$ cells (if $n >> p$) and at least $2^{1-o(1))\cdot n}p^{n-1}$ cells (if $n << p$).

We take the product of this cellulation with itself to get a cellulation of $S^n \times S^n$. The degrees of freedom will be on the q-cells for $q = n$, so that N is again exponential in n. We have $\log_2(N) = 2(1+\log_2(p)+o(1))\cdot n$ for $n >> p$ and $\log_2(N) = 2(1+\log_2(p)+o(1))\cdot n - 2\log_2(p)$ for $n << p$. We will take $p = 1/\log(N), n = \Theta(\log(N)/\log(\log(N)))$.

Each qubit is acted on by $2n$ stabilizers and each stabilizer acts on $2(n+1)$ qubits. Hence, the weight is logarithmic in N, $w = \Theta(\log(N)/\log(\log(N)))$.

The number of encoded qubits is equal to 2, as can be computed from the homology of $S^n \times S^n$ (by the Künneth formula, $H_i(S^n \times S^n; \mathbb{Z}_2) = 2$ for $i = n$, $H_i(S^n \times S^n; \mathbb{Z}_2) = 1$ for $i = 0, 2n$, and $H_n(S^n \times S^n; \mathbb{Z}_2) = 0$ otherwise).

▶ **Lemma 32.** *For the hypersphere product code,*

$$d_X(w) = d_Z(w) = (p+1)^{n+1}\Big(1-(\frac{p+1-2}{p+1})^{n+1}\Big) = \Theta(N^{\frac{p}{p+2}}). \tag{37}$$

For $p = \Omega(\log(N))$,

$$d_X(w) = d_Z(w) = \Theta(\sqrt{N})). \tag{38}$$

Proof. Let a be a 0-cell in the second S^n in the product $S^n \times S^n$. Let $Z(a,2)$ be the logical Z operator which is the product Z_i over all i which are the product of an n-cell in the first S^n with 0-cell a. Then, any logical X operator which anticommites with $Z(a,2)$ must have some support on some cell i which is a product of an n-cell in the first S^n with 0-cell a. However, since $Z(a,2)$ and $Z(b,2)$ are homologous for any two choices of 0-cells a, b in the second S^n ($Z(a,2), Z(b,2)$ differ by a product of stabilizers), that logical X operator must have some support on some cell i which is a product of an n-cell in the first S^n with 0-cell a for *all* 0-cells a in the second S^n. Hence, that logical X operator must have a number of cells in its support equal to the number of 0-cells in the second S^n. This number is equal to $(p+1)^{n+1}\Big(1-(\frac{p+1-2}{p+1})^{n+1}\Big)$.

This number is also an upper bound to $d_X(w)$, since the product of X over all cells which are a product of a fixed n-cell in the first S^n with an arbitrary 0-cells in the second S^n is a logical operator.

We can similarly lower bound the number of cells in the support of any logical X operator which anticommutes with the operator $Z(a,1)$, defined to be the logical Z operator which is the product Z_i over all i which are the product of an n-cell in the *second* S^n with 0-cell a in the *first* S^n. ◀

We now show soundness. Again, the geometric interpretation is to shrink the boundary to a point.

▶ **Lemma 33.** *For the hypersphere product code,* $\epsilon_X(w) = \epsilon_Z(w) \geq \Omega(1/\log(N)^2)$.

Proof. Consider any $v \in \mathcal{C}_q$ with $\partial_q v \neq 0$.

We place coordinates $[0,p]^{n+1}$ on the first large hypercube. Call the face where the first coordinate is equal to p the "top face". Call the face where the first coordinate is equal to 0 the "bottom face". Let $v_0 = v$. We will construct a sequence $v_1, v_2, \ldots, v_f \in \mathcal{C}_q$ for some integer f where we bound $\mathrm{wt}(v_{i+1} - v_i)$ with the final vector $v_f = 0$. In this way, we will bound $\mathrm{wt}(v_0)$. We construct the sequence so that the boundaries $\partial_q v_i$ are first removed from the top face of the first hypercube, then moved from the top face to the bottom face of the first hypercube, and finally moved to a point on the bottom face of the first hypercube.

Throughout this proof, when we refer to coordinates, we refer to the first hypercube in the product. We regard an r-cell as being a product of 0-cells and 1-cells. That is, each cell in the product of hypercubes is product of cells in each hypercube. Then, each r-cell in a hypercube is a product of r 1-cells and $n+1-r$ 0-cells. The $n+1$ different terms in the product correspond to different coordinates. When we say that a cell "is a 0-cell" in a given coordinate, we mean that the cell is a product of a 0-cell in that coordinate with some cells in other coordinates.

We first explain the middle step, moving from top face to bottom face. Suppose that some v_i has $\partial_q v_i$ vanishing on the top face. Indeed, suppose that $\partial_q v_i$ vanishes if the first coordinate is greater than x, for some integer x. Then, let $\pi_x(\partial_q v_i)$ be the projection of $\partial_q v_i$ onto cells with first coordinate equal to x. This projection consists only of cells which are 0-cells in the first coordinate. Let $v_{i+1} - v_i$ be defined by taking $\pi_x(\partial_q i)$ and replacing every 0-cell in the first coordinate at position x with a 1-cell at position $x - 1/2$. Then, $\pi_x(\partial_q v_{i+1}) = 0$. Iterating this procedure, decreasing x from p, to $p-1$, to $p-2$, and so on, we can construct a sequence $v_i, v_{i+1}, \ldots$ so that the final vector in the second has boundary only on the bottom face. There are at most p steps in the sequence. Note that because $\partial^2 = 0$, once we ensure that $\pi_x(\partial_q v_i) = 0$, then we know that $\partial_q v_i$ has not cells which are 1-cell in the first coordinate with midpoint as $x - 1/2$.

Now we explain the first step, moving the boundary off the top face. We apply above the above procedure to the *second* coordinate. We let $\pi_{p,x}(\partial_q v_i)$ be the projection $\partial_q v_i$ onto cells with first coordinate equal to p and second coordinate equal to x, for integer x. We then construct a sequence so that this projection vanishes for $x = p, p-1, \ldots$, following the same procedure as in the above paragraph. There are at most p steps in this sequence. We then repeat this for the second coordinate, third coordinate, and so on; giving at most pn steps.

The final step is the same as the first step, with the top face replaced by the bottom face.

So, there are at most $\mathcal{O}(pn)$ steps in the sequence. We have $\mathrm{wt}(\partial_q v_i) \leq \mathrm{wt}(\partial_q v)$ for all vectors in the sequence, and there are at most $\mathcal{O}(pn)$ steps, so this gives $\epsilon_X(w) \geq \Omega(1/pn) = \Omega(1/\log(N)^2)$. ◀

In the above construction, we lost a factor of n due to having n steps in the sequence to move the boundary. Likely for this construction, this factor cannot be avoided since the diameter of the hypercube is pn. One might wonder whether other geometries (such as a geometry that more closely approximates a sphere) would improve on this factor; note however that since the volume of a sphere of radius r in n-dimensional Euclidean space scales roughly as $(r/n)^{n/2}$, one would need to take the radius proportional to n in order to obtain a large volume so again one would need to have a large diameter for the geometry.

7 Discussion

We have presented several different code constructions based on the toric code on families of higher-dimensional manifolds. Rather than varying the geometry or topology at fixed dimension, as is more commonly done, we have considered varying dimension. This leads to a scaling in which the number of qubits, N, scales exponentially with dimension, n, so that the weight of the stabilizers w is proportional $n \propto \log(N)$. Assuming conjecture 1, we have constructed a code family with almost linear distance and logarithmic weight.

Acknowledgments. I thank L. Eldar and M. Freedman for useful discussions.

References

1 Dorit Aharonov and Lior Eldar. Quantum locally testable codes. *SIAM Journal on Computing*, 44(5):1230–1262, 2015.

2 Ivan Babenko and Mikhail Katz. Systolic freedom of orientable manifolds. *Annales scientifiques de l'Ecole normale supérieure*, 31(6):787–809, 1998. doi:10.1016/S0012-9593(99)80003-2.

3 Dave Bacon, Steven T Flammia, Aram W Harrow, and Jonathan Shi. Sparse quantum codes from quantum circuits. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 327–334. ACM, 2015.

4 H. Bombin and M. A. Martin-Delgado. Homological error correction: Classical and quantum codes. *Journal of Mathematical Physics*, 48(5):052105, may 2007. doi:10.1063/1.2731356.

5 Sergey Bravyi and Matthew B Hastings. Homological product codes. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 273–282. ACM, 2014.

6 A. R. Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. *Physical Review A*, 54(2):1098–1105, aug 1996. doi:10.1103/physreva.54.1098.

7 John Horton Conway and Neil James Alexander Sloane. *Sphere packings, lattices and groups*, volume 290. Springer Science & Business Media, 2013.

8 Renaud Coulangeon. Minimal vectors in the second exterior power of a lattice. *Journal of algebra*, 194(2):467–476, 1997.

9 Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. Topological quantum memory. *Journal of Mathematical Physics*, 43(9):4452–4505, 2002.

10 Uri Erez, Simon Litsyn, and Ram Zamir. Lattices which are good for (almost) everything. *IEEE Transactions on Information Theory*, 51(10):3401–3416, 2005.

11 Herbert Federer and Wendell H Fleming. Normal and integral currents. *Annals of Mathematics*, 72:458–520, 1960.

12 Federer-Fleming deformation theorem. Encyclopedia of Mathematics. URL: http://www.encyclopediaofmath.org/index.php?title=Federer-Fleming_deformation_theorem&oldid=28190.

13 Michael H. Freedman and David A. Meyer. Projective plane and planar quantum codes. *Foundations of Computational Mathematics*, 1(3):325–332, jul 2001. doi:10.1007/s102080010013.

14 Michael H Freedman, David A Meyer, and Feng Luo. Z2-systolic freedom and quantum codes. *Mathematics of quantum computation, Chapman & Hall/CRC*, pages 287–320, 2002.

15 Robert Gallager. Low-density parity-check codes. *IRE Transactions on information theory*, 8(1):21–28, 1962.

16 Nicolas Gama, Nick Howgrave-Graham, Henrik Koy, and Phong Q Nguyen. Rankin?s constant and blockwise lattice reduction. In *Annual International Cryptology Conference*, pages 112–130. Springer, 2006.

17 Reese Harvey and H Blaine Lawson. Calibrated geometries. *Acta Mathematica*, 148(1):47–157, 1982.

18 A.Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, jan 2003. doi:10.1016/s0003-4916(02)00018-0.

19 Anthony Leverrier, Jean-Pierre Tillich, and Gilles Zémor. Quantum expander codes. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 810–824. IEEE, 2015.

20 David JC MacKay. Good error-correcting codes based on very sparse matrices. *IEEE transactions on Information Theory*, 45(2):399–431, 1999.

21 Jacques Martinet. *Perfect lattices in Euclidean spaces*, volume 327. Springer Science & Business Media, 2013.

22 John Willard Milnor and Dale Husemoller. *Symmetric bilinear forms*, volume 60. Springer, 1973.

23 David Poulin. Stabilizer formalism for operator quantum error correction. *Physical review letters*, 95(23):230504, 2005.

24 Robert Alexander Rankin. On positive definite quadratic forms. *Journal of the London Mathematical Society*, 1(3):309–314, 1953.

25 Wolfgang M Schmidt et al. Asymptotic formulae for point lattices of bounded determinant and subspaces of bounded height. *Duke Mathematical Journal*, 35(2):327–339, 1968.

26 Jean-Pierre Tillich and Gilles Zémor. Quantum ldpc codes with positive rate and minimum distance proportional to $n^{1/2}$. *arXiv preprint arXiv:0903.0566*, 2009.

Conditional Hardness for Sensitivity Problems

Monika Henzinger[*1], **Andrea Lincoln**[†2], **Stefan Neumann**[3], **and Virginia Vassilevska Williams**[‡4]

1 **University of Vienna, Faculty of Computer Science, Vienna, Austria**
monika.henzinger@univie.ac.at
2 **Stanford University, Computer Science Department, Stanford, USA**
andreali@cs.stanford.edu
3 **University of Vienna, Faculty of Computer Science, Vienna, Austria**
stefan.neumann@univie.ac.at
4 **Stanford University, Computer Science Department, Stanford, USA**
virgi@cs.stanford.edu

Abstract

In recent years it has become popular to study dynamic problems in a sensitivity setting: Instead of allowing for an arbitrary sequence of updates, the sensitivity model only allows to apply batch updates of small size to the *original* input data. The sensitivity model is particularly appealing since recent strong conditional lower bounds ruled out fast algorithms for many dynamic problems, such as shortest paths, reachability, or subgraph connectivity.

In this paper we prove conditional lower bounds for these and additional problems in a sensitivity setting. For example, we show that under the Boolean Matrix Multiplication (BMM) conjecture combinatorial algorithms cannot compute the $(4/3-\varepsilon)$-approximate diameter of an undirected unweighted dense graph with truly subcubic preprocessing time and truly subquadratic update/query time. This result is surprising since in the static setting it is not clear whether a reduction from BMM to diameter is possible. We further show under the BMM conjecture that many problems, such as reachability or approximate shortest paths, cannot be solved faster than by recomputation from scratch even after *only one or two* edge insertions. We extend our reduction from BMM to Diameter to give a reduction from All Pairs Shortest Paths to Diameter under one deletion in weighted graphs. This is intriguing, as in the static setting it is a big open problem whether Diameter is as hard as APSP. We further get a nearly tight lower bound for shortest paths after two edge deletions based on the APSP conjecture. We give more lower bounds under the Strong Exponential Time Hypothesis. Many of our lower bounds also hold for static oracle data structures where no sensitivity is required.

Finally, we give the first algorithm for the $(1+\varepsilon)$-approximate radius, diameter, and eccentricity problems in directed or undirected unweighted graphs in case of single edges failures. The algorithm has a truly subcubic running time for graphs with a truly subquadratic number of edges; it is tight w.r.t. the conditional lower bounds we obtain.

1998 ACM Subject Classification F.2.2 Computations on discrete structures

Keywords and phrases sensitivity, conditional lower bounds, data structures, dynamic graph algorithms

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.26

[*] The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement no. 340506.
[†] Supported by a Stanford Graduate Fellowship.
[‡] VVW and AL were supported by NSF Grants CCF-1417238, CCF-1528078 and CCF-1514339, and BSF Grant BSF:2012338.

© Monika Henzinger, Andrea Lincoln, Stefan Neumann, and Virginia Vassilevska Williams;
licensed under Creative Commons License CC-BY

1 Introduction

A dynamic algorithm is an algorithm that is able to handle changes in the input data: It is given an input instance x and is required to maintain certain properties of x while x undergoes (possibly very many) updates. For example, an algorithm might maintain a graph, which undergoes edge insertions and deletions, and a query is supposed to the return the diameter of the graph after the updates. Often dynamic algorithms are also referred to as data structures. During the last few years strong conditional lower bounds for many dynamic problems were derived (see, e.g., [36, 3, 27, 4, 21, 1, 32]), which rule out better algorithms than simple recomputation from scratch after each update or before each query.

Partially due to this, in recent years it has become popular to study dynamic problems in a more restricted setting that only allows for a *bounded* number of changes to the input instance (see, for example, [37, 23, 10, 19], and the references in Table 4). These algorithms are usually referred to as *sensitivity*[1] data structures. The hope is to obtain algorithms in the sensitivity setting which are faster than the conditional lower bounds for the general setting.

More formally, a *data structure with sensitivity* d for a problem P has the following properties: It obtains an instance p of P and is allowed polynomial preprocessing time on p. After the preprocessing, the data structure must provide the following operations:

(Batch) Update. Up to d changes are performed to the *initial* problem instance p, e.g., d edges are added to or removed from p.

Query. The user queries a specific property about the instance of the problem after the last update, e.g., the shortest path between two nodes avoiding the edges deleted in the last update.

The parameter d bounding the batch update size is referred to as the *sensitivity* of the data structure. Note that every batch update is performed on the *original* problem instance. Thus, in contrast to "classic" dynamic algorithms (without sensitivity), a query only reflects the changes made to p by the *last* batch update and *not* by previous batch updates. As the size of a batch update is constrained to at most d, each query is executed on a graph that differs from p by at most d edges. After a batch update an arbitrary number of queries may be performed.

Some data structures (usually called *oracles*) combine a query and an update into a single operation, i.e., the combined operation obtains an input tuple (Q, U), where Q is a query and U is an update. A special case are *static oracles*, which have $U = \emptyset$. The conditional lower bounds we derive in this paper also hold in this setting, since oracles with an efficient combined operation can be used to solve sensitivity problems.

While some existing sensitivity data structures can preprocess the answers to all possible updates and queries during their preprocessing time, this is not possible in general (due to constraints in the preprocessing time and the fact that the number of possible updates/queries grows exponentially in the parameter d). Hence, we still consider a sensitivity data structure a dynamic (instead of static) algorithm.

[1] Sometimes sensitivity data structures are also called "fault-tolerant" or "emergency planning" algorithms. See Appendix A.2 for a discussion of terminology.

The Hypotheses

We state the hypotheses on which we base the conditional lower bounds in this paper. By now they are all considered standard in proving fine-grained reduction-based lower bounds. For a more detailed description of the hypotheses, see, e.g., Abboud and Williams [3] and the references therein. As usual we work in the word-RAM model of computation with word length of $O(\log n)$ bits. The hypotheses below concern the complexity of the Boolean Matrix Multiplication (BMM), Satisfiability of Boolean Formulas in Conjunctive Normal Form (CNF-SAT), All Pairs Shortest Paths (APSP), Triangle Detection and Online Boolean Matrix Vector Multiplication (OMv) problems. Other popular hypotheses from prior work consider other famous problems such as 3SUM and other sparsity regimes such as triangle detection in very sparse graphs (see, e.g. [3]).

▶ **Conjecture 1** (Impagliazzo, Paturi and Zane [29, 30]). *The Strong Exponential Time Hypothesis (SETH) states that for each $\varepsilon > 0$, there exists a $k \in \mathbb{N}$, such that k-SAT cannot be solved in time $O(2^{n(1-\varepsilon)} \operatorname{poly}(n))$.*

▶ **Conjecture 2.** *The Boolean Matrix Multiplication (BMM) conjecture states that for all $\varepsilon > 0$, there exists no* combinatorial *algorithm that computes the product of two $n \times n$ matrices in expected time $O(n^{3-\varepsilon})$.*

Note that BMM can be solved in truly subcubic using fast matrix multiplication (FMM): the current fastest algorithms run in $O(n^{2.373})$ time [41, 25]. However, algorithms using FMM are not considered to be combinatorial. Formally, the term *combinatorial* algorithm is not well-defined and it is common to rule out the use of FMM or other "Strassen-like" methods in the design of such algorithms as most of them are not considered practical. True combinatorial algorithms are not only considered practical but also easily extendable. For instance, prior work on combinatorial BMM algorithms has almost always led to an algorithm for APSP with similar running time (e.g. [5] and [17]).

One of the simplest graph problems is that of detecting whether the graph contains a triangle, i.e., three nodes with all three edges between them. Itai and Rodeh [31] showed that any algorithm for BMM can solve Triangle detection in the same time. Conversely, Vassilevska Williams and Williams [42] showed that any truly subcubic combinatorial algorithm for Triangle Detection can be converted into a truly subcubic combinatorial algorithm for BMM. Hence, the BMM conjecture implies there is no truly subcubic *combinatorial* algorithm for Triangle Detection. We use this fact and the resulting Triangle Conjecture that there is no truly subcubic algorithm for Triangle Detection in our reductions based on BMM.

The following is a popular conjecture about the APSP problem.

▶ **Conjecture 3.** *The APSP conjecture states that given a graph G with n vertices, m edges, and edge weights in $\{1, \ldots, n^c\}$ for some constant c, the All Pairs Shortest Paths problem (APSP) cannot be solved in $O(n^{3-\varepsilon})$ expected time for any $\varepsilon > 0$.*

Similar to the relationship between BMM and Triangle Detection, [42] showed that there is a triangle problem in weighted graphs, Negative Triangle, that is equivalent under subcubic reductions to APSP. We use that problem in our reductions.

Our final conjecture concerns Boolean matrix vector product.

▶ **Conjecture 4** (Henzinger et al. [27]). *Let B be a Boolean matrix of size $n \times n$. In the Online Matrix-vector (OMv) problem, n binary vectors $v_1, \ldots, v_n$ of size n appear online and an algorithm solving the problem must output the vector Bv_i before the next vector v_{i+1} arrives.*

The OMv conjecture states that for all $\varepsilon > 0$ and after any polynomial time preprocessing of B, it takes $\Omega(n^{3-\varepsilon})$ time to solve the OMv problem with error probability at most $1/3$.

Most of the conjectures are stated w.r.t. *expected* time, i.e., the conjectures rule out randomized algorithms. In case of dynamic algorithms using randomness, it is common to argue if an oblivious or a non-oblivious adversary is allowed. Previous literature on conditional lower bounds for dynamic algorithms did not explicitly state what kind of adversaries are allowed for their lower bounds. We give a quick discussion of this topic in Appendix A.3.

Our Results

In this paper we develop a better understanding of the possibilities and limitations of the sensitivity setting by providing conditional lower bounds for sensitivity problems. We show that under plausible assumptions for many dynamic graph problems even the update of only *one or two* edges cannot be solved faster than by re-computation from scratch. See Table 2 and Table 3 in the Appendix for a list of all our conditional lower bounds for sensitivity data structures, and our lower bounds for static oracles respectively. Table 1 gives explanations of the problems. The abbreviations used in the tables are explained in its captions. We next discuss our main results.

New reductions

We give several new reductions for data structures with small sensitivity.

(1) We give a novel reduction from triangle detection and BMM to maintaining an approximation of the diameter of the graph and eccentricities of all vertices, under a single edge failure. This is particularly surprising because in the static case it is unknown how to reduce BMM to diameter computation. Using the BMM conjecture this results in lower bounds of $n^{3-o(1)}$ on the preprocessing time or of $n^{2-o(1)}$ update or query time for $(4/3-\varepsilon)$-approximate decremental diameter and eccentricity in unweighted graphs *with sensitivity 1*, i.e., when a single edge is deleted. Those results are tight w.r.t. the algorithm we present in Section 5.

(2) A particular strength of BMM-based reductions is that they can very often be converted into APSP-based lower bounds for weighted variants of the problems. APSP-based lower bounds, in turn, no longer require the "combinatorial"-condition on the algorithms, making the lower bounds stronger. We show how our BMM-based lower bounds for approximate diameter with sensitivity 1 can be converted into an APSP-based lower bound for diameter with sensitivity 1 in weighted graphs. In particular, we show that unless APSP has a truly subcubic algorithm, any data structure that can support diameter queries for a single edge deletion must either have essentially cubic preprocessing time, or essentially quadratic query time. This lower bound is tight w.r.t. to a trivial algorithm using the data structure of [10]. The APSP to 1-sensitive Diameter lower bound is significant also because it is a big open problem whether in the static case Diameter and APSP are actually subcubically equivalent (see e.g. [2]).

(3) We consider the problem of maintaining the distance between two fixed nodes s and t in an undirected weighted graph under edge failures. The case of a *single edge failure* can be solved in m edge, n node graphs with essentially optimal $O(m\alpha(n))$ preprocessing time and $O(1)$ query time with an algorithm of Nardelli et al. [33]. The case of two edge failures has been open for some time. We give compelling reasons for this by showing that under the APSP conjecture, maintaining the s-t distance in an unweighted graph under two edge failures requires either $n^{3-o(1)}$ preprocessing time or $n^{2-o(1)}$ query time. Notice that with no preprocessing time, just by using Dijkstra's algorithm at query time, one can obtain $O(n^2)$ query time. Similarly, one can achieve $\tilde{O}(n^3)$ preprocessing time and $O(1)$

query time by applying the single edge failure algorithm of [33] n times at preprocessing, once for $G \setminus \{e\}$ for every e on the shortest st path. Thus our lower bound shows that under the APSP conjecture, the naive recomputation time is essentially optimal.

(4) We show lower bounds with sensitivity d for deletions-only and insertions-only $(2-\varepsilon)$-approximate single source and $(5/3-\varepsilon)$-approximate st-shortest paths in undirected unweighted graphs, as well as for weighted bipartite matching problems under the OMv conjecture. The lower bounds show that with polynomial in n preprocessing either the update time must be super-polynomial in d or the query time must be $d^{1-o(1)}$.

New upper bounds

We complement our lower bounds with an algorithm showing that some of our lower bounds are tight: In particular, we present a deterministic combinatorial algorithm that can compute a $(1+\varepsilon)$-approximation (for any $\varepsilon > 0$) for the eccentricity of any given vertex, the radius and the diameter of a directed or undirected unweighted graph after single edge failures. The preprocessing time of the data structure is $\tilde{O}(mn + n^{1.5}\sqrt{Dm/\varepsilon})$, where D is the diameter of the graph and m and n are the number of edges and vertices; the query time is constant. Since $D \leq n$, the data structure can be preprocessed in time $\tilde{O}(n^2\sqrt{m/\varepsilon})$. In particular, for sparse graphs with $m = \tilde{O}(n)$, it takes time $\tilde{O}(n^{2.5}\varepsilon^{-\frac{1}{2}})$ to build the data structure. Our lower bounds from BMM state that even getting a $(4/3-\varepsilon)$-approximation for diameter or eccentricity after a *single* edge deletion requires either $n^{3-o(1)}$ preprocessing time, or $n^{2-o(1)}$ query or update time. Hence, our algorithm's preprocessing time is tight (under the conjecture) since it has constant time queries.

Conditional Lower Bounds based on modifications of prior reductions

Some reductions in prior work [42, 3] only perform very few updates before a query is performed or they can be modified to do so. After the query, the updates are "undone" by rolling back to the initial instance of the input problem. Hence, some of their reductions also hold in a sensitivity setting. Specifically we achieve the following results in this way:

(1) Based on the BMM conjecture we show that for reachability problems with st-queries already *two* edge insertions require $n^{3-o(1)}$ preprocessing time or $n^{2-o(1)}$ update or query time; for ss-queries we obtain the same bounds even for a *single* edge insertion. This lower bound is matched by an algorithm that recomputes at each step.

(2) We present strong conditional lower bounds for static oracle data structures. We show that under the BMM conjecture, oracle data structures that answer about the reachability between any two queried vertices cannot have truly subcubic preprocessing time *and* truly subquadratic query time. This implies that combinatorial algorithms *either* essentially need to compute the transitive closure matrix of the graph during the preprocessing time *or* essentially need to traverse the graph at each query. We show the same lower bounds for static oracles that solve the $(5/3-\varepsilon)$-approximate ap-shortest paths problem in undirected unweighted graphs. This shows that we essentially cannot do better than solving APSP in the preprocessing or computing the distance in each query.

(3) The subcubic equivalence between the replacement paths problem and APSP [42] immediately leads to a conditional lower bound for s-t distance queries with sensitivity 1 in *directed*, weighted graphs. Our lower bound for s-t distance queries with sensitivity 2 in undirected graphs is inspired by this reduction. The lower bound for sensitivity 1 is matched by the algorithm of Bernstein and Karger [10].

Similarly, a reduction from BMM to replacement paths in directed unweighted graphs from [42] shows that the $O(m\sqrt{n})$ time algorithm of Roditty and Zwick [38] is optimal among all combinatorial algorithms, for every choice of m as a function of n. It also immediately implies that under the BMM conjecture, combinatorial s-t distance 1-sensitivity oracles in unweighted graphs require either $mn^{0.5-o(1)}$ preprocessing time or $m/n^{0.5+o(1)}$ query time, for every choice of m as a function of n; this is tight due to Roditty and Zwick's algorithm. (The combinatorial restriction is important here as there is a faster $\tilde{O}(n^{2.373})$ time non-combinatorial algorithm for replacement paths [40] and hence for distance sensitivity oracles in directed unweighted graphs.)

(4) We additionally provide new lower bounds under SETH: We show that assuming SETH the #SSR problem cannot be solved with truly subquadratic update and query times when any constant number of edge insertions is allowed; this matches the lower bound for the general dynamic setting. For the ST-reachability problem and the computation of $(4/3-\varepsilon)$-approximate diameter we show that under SETH truly sublinear update and query times are not possible even when only a constant number of edge insertions are supported. The sensitivity of the reductions is a constant $K(\varepsilon,t)$ that is determined by the preprocessing time $O(n^t)$ we allow and some properties of the sparsification lemma [30]. Notice that while the constant $K(\varepsilon,t)$ depends on the preprocessing time and the constant in the sparsification lemma, it does *not* depend on any property of the SAT instance in the reduction. See Section 4 for a thorough discussion of the parameter $K(\varepsilon,t)$. The lower bound for #SSR shows that we cannot do better than recomputation after each update.

(5) Using a reduction from OMv we show lower bounds with sensitivity d for deletions-only or insertions-only st-reachability, strong connectivity in directed graphs. The lower bounds show that with polynomial in n preprocessing either the update time must be super-polynomial d or the query time must be $\Omega(d^{1-\varepsilon})$.

Related Work

In the last few years many conditional lower bounds were derived for dynamic algorithms. Abboud and Williams [3] gave such lower bounds under several different conjectures. New lower bounds were given by Henzinger et al. [27], who introduced the OMv conjecture, and by Abboud, Williams and Yu [4], who stated combined conjectures that hold as long as either the 3SUM conjecture *or* SETH *or* the APSP conjecture is correct. Dahlgaard [21] gave novel lower bounds for partially dynamic algorithms. Abboud and Dahlgaard [1] showed the first hardness results for dynamic algorithms on planar graphs and Kopelowitz, Pettie and Porat [32] gave stronger lower bounds from the 3SUM conjecture. However, none of the lower bounds mentioned in the above papers explicitly handled the sensitivity setting.

During the last decade there have been many new algorithms designed for the sensitivity setting. In Section A.5 we give a short discussion summarizing many existing algorithms.

2 Lower Bounds From Boolean Matrix Multiplication

The following theorem summarizes the lower bounds we derived from the BMM conjecture.

▶ **Theorem 5.** *Assuming the BMM conjecture, combinatorial algorithms cannot solve the following problems with preprocessing time $O(n^{3-\varepsilon})$, and update and query times $O(n^{2-\varepsilon})$ for any $\varepsilon > 0$:*

1. *incremental st-reachability with sensitivity 2,*

2. *incremental ss-reachability with sensitivity 1,*
3. *static ap-reachability,*
4. *$(7/5-\varepsilon)$-approximate st shortest paths in undirected unweighted graphs with sensitivity 2,*
5. *$(3/2-\varepsilon)$-approximate ss shortest paths in undirected unweighted graphs with sensitivity 1,*
6. *static $(5/3-\varepsilon)$-approximate ap shortest paths*
7. *decremental $(4/3-\varepsilon)$-approx. diameter in undirected unweighted graphs with sensitivity 1,*
8. *decremental $(4/3-\varepsilon)$-approx. eccentricity in undirected unweighted graphs with sensitivity 1.*

Additionally, under the BMM conjecture, decremental st-shortest paths with sensitivity 1 *in directed unweighted graphs with n vertices and $m \geq n$ edges require either $m^{1-o(1)}\sqrt{n}$ preprocessing time or $m^{1-o(1)}/\sqrt{n}$ query time for every function m of n.*

A strength of the reductions from BMM is that they can usually be extended to provide APSP-based reductions for weighted problems without the restriction to combinatorial algorithms; we do this in Section 3. While we state our results in the theorem only for combinatorial algorithms under the BMM conjecture, we would like to point out that they also hold *for any kind of algorithm* under a popular version of the triangle detection conjecture for sparse graphs that states that finding a triangle in an m-edge graph requires $m^{1+\delta-o(1)}$ time for some $\delta > 0$. Our lower bounds then rule out algorithms with a preprocessing time of $O(m^{1+\delta-\varepsilon})$ and update and query times $O(m^{2\delta-\varepsilon})$ for any $\varepsilon > 0$.

Many of the bullets of the theorem follow from prior work via a few observations, which we discuss in Appendix A.6. Our results on decremental diameter and eccentricity, however, are completely novel. In fact, it was completely unclear before this work whether such results are possible. Impagliazzo et al. [16] define a strengthening of SETH under which there can be no deterministic fine-grained reduction from problems such as APSP and BMM to problems such as orthogonal vectors or diameter in sparse graphs. It is not clear whether a reduction from BMM to diameter in dense graphs is possible, as the same "quantifier issues" that arise in the sparse graph case arise in the dense graph case as well: Diameter is an $\exists\forall$-type problem (i.e., do there exist two nodes such that all paths between them are long?), and BMM is equivalent to Triangle detection which is an $\exists$-type problem (i.e., do there exist three nodes that form a clique?).

Decremental Diameter

We give the reduction from BMM to decremental diameter in undirected unweighted graphs with sensitivity 1. Note that the lower bound also holds for eccentricity oracles: Instead of querying the diameter n times, we can query the eccentricity of a variable vertex n times.

Let $G = (V, E)$ be an undirected unweighted graph for Triangle Detection. We construct a graph G' as follows.

We create four copies of V denoted by V_1, V_2, V_3, V_4, and for $i = 1, 2, 3$, we add edges between nodes $u_i \in V_i$ and $v_{i+1} \in V_{i+1}$ if $(u, v) \in E$. We create vertices a_v and b_v for each $v \in V$, and denote the set of all a_v by A and the set of all b_v by B. We connect the vertices in A to a clique and also those of B. For each $v \in V$, we add an edge (v_1, a_v) and an edge (a_v, b_v). A node b_v is connected to all vertices in V_4. We further introduce two additional vertices c, d, which are connected by an edge. We add edges between c and all nodes in V_2 and V_3, and between d and all nodes in V_3 and V_4. The node c has an edge to each vertex in A and the node d has an edge to each vertex in B. Notice that the resulting graph has $O(n)$ vertices and $O(n^2)$ edges. We visualized the graph in Figure 2 in the appendix.

Note that even without the edges from $B \times V_4$, no pair of nodes has distance larger than 3, except for pairs of nodes from $V_1 \times V_4$. If a node v participates in a triangle in G, then in

G' there is a path of length 3 from v_1 to v_4 without an edge from $B \times V_4$. Otherwise, there is no such path, i.e., the diameter increases to 4 after the deletion of (b_v, v_4).

We perform one stage per vertex $v \in V$: Consider the copy $v_4 \in V_4$ of v. We remove the edge (b_v, v_4) and query the diameter of the graph. We claim that G has a triangle iff one of the queries returns diameter 3.

▶ **Lemma 6.** *For each vertex v in G, the diameter of $G' \setminus \{(b_v, v_4)\}$ is larger than 3 if and only if v does not participate in a triangle in G.*

Proof. Assume that G has a triangle $(v, u, w) \in V^3$ and consider the stage for v. Notice that only the shortest paths change that used edge (b_v, v_4); this is not the case for any $z \neq v$, because the path $z_1 \to a_z \to b_z \to z_4$ is not affected by the edge deletion. We only need to consider the path $v_1 \to a_v \to b_v \to v_4$. Since G has a triangle (v, u, w), there exists the path $v_1 \to u_2 \to w_3 \to v_4$ of length 3 as desired. Hence, the diameter is 3.

Assume the query in the stage for vertex $v \in V$ returned diameter 3. Since we deleted the edge (b_v, v_4), there is no path of length 3 from v_1 to v_4 via A and B. Hence, the new shortest path from v_1 to v_4 must have the form $v_1 \to u_2 \to w_3 \to v_4$. By construction of the graph, this implies that G has a triangle (v, u, w). ◀

Altogether we perform n queries and n updates. Thus under the BMM conjecture any combinatorial algorithm requires $n^{3-o(1)}$ preprocessing time or $n^{2-o(1)}$ update or query time.

3 Sensitivity Lower Bounds from the APSP Conjecture

In this section we present new lower bounds based on the APSP conjecture. These lower bounds hold for arbitrary, not necessarily combinatorial, algorithms. We present our results in the following theorem and give the proofs in Appendix A.7.

▶ **Theorem 7.** *Assuming the APSP conjecture, no algorithms can solve the following problems with preprocessing time $O(n^{3-\varepsilon})$, and update and query times $O(n^{2-\varepsilon})$ for any $\varepsilon > 0$:*

1. *Decremental st-shortest paths in directed weighted graphs with sensitivity* 1,
2. *decremental st-shortest paths in undirected weighted with sensitivity* 2,
3. *decremental diameter in undirected weighted graphs with sensitivity* 1.

Decremental *st*-shortest paths in directed weighted graphs with sensitivity 1

In 2010, Vassilevska Williams and Williams [42] showed that the so called Replacement Paths (RP) problem is subcubically equivalent to APSP. RP is defined as follows: given a directed weighted graph G and two nodes s and t, compute for every edge e in G, the distance between s and t in $G \setminus \{e\}$. Note that only the deletion of the at most $n-1$ edges on the shortest path from s to t affect the distance from s to t. This has an immediate implication for 1-sensitivity oracles for st-shortest paths: The APSP conjecture would be violated by any 1-sensitivity oracle that uses $O(n^{3-\varepsilon})$ preprocessing time and can answer distance queries between two fixed nodes s and t with one edge deletion in time $O(n^{2-\varepsilon})$ for any $\varepsilon > 0$.

Decremental *st*-shortest paths in undirected weighted with sensitivity 2

With this we are able to show that on *undirected* weighted graphs finding a shortest path between fixed s and t with 2 edge deletions cannot be done with truly sub-cubic preprocessing time and truly subquadratic query time assuming the APSP conjecture. This is surprising

because in the case of a *single edge failure* Nardelli et al. [33] show that shortest paths can be solved with an essentially optimal $O(m\alpha(n))$ preprocessing time and $O(1)$ query time. Thus, assuming the APSP conjecture we show a separation between 1 sensitivity and 2 sensitivity. Additionally, with sensitivity 2 and no preprocessing time $O(n^2)$ update time is achievable, and with $\tilde{O}(n^3)$ preprocessing time using Nardelli et al. we can get an $O(1)$ query time. Thus, we show these approaches are essentially tight assuming the APSP conjecture. The full reduction is in Appendix 3.

Decremental diameter in undirected weighted graphs with sensitivity 1

A nice property of BMM-based reductions is that they can very often be converted to APSP-based reductions to weighted versions of problems. Here we convert our BMM-based reduction for decremental 1-sensitive Diameter to a reduction from APSP into decremental 1-sensitive diameter in undirected weighted graphs. Note that, as in the BMM case we can get the same lower bounds for eccentricity.

4 SETH Lower Bounds with Constant Sensitivity

In this section, we prove conditional lower bounds with constant sensitivity from SETH. Before we give the reductions, we first argue about what their sensitivities are.

▶ **Theorem 8.** *Let $\varepsilon > 0$, $t \in \mathbb{N}$. The SETH implies that there exists no algorithm with preprocessing time $O(n^t)$, update time $u(n)$ and query time $q(n)$, such that $\max\{u(n), q(n)\} = O(n^{1-\varepsilon})$ for the following problems:*

1. *Incremental #SSR with constant sensitivity $K(\varepsilon, t)$,*
2. *$(4/3-\varepsilon)$-approximate incremental diameter with constant sensitivity $K(\varepsilon, t)$,*
3. *incremental ST-Reach with constant sensitivity $K(\varepsilon, t)$.*

We prove the theorem in Appendix A.8. The parameter $K(\varepsilon, t)$ is explained in the following paragraph*.

The Sensitivity of the Reductions

The conditional lower bounds we prove from SETH hold even for constant sensitivity; however, the derivation of these constants is somewhat unnatural. Nonetheless, we stress that our lower bounds hold for constant sensitivity and in particular for every algorithm with sensitivity $\omega(1)$.

To derive the sensitivity of our reductions, we use a similar approach as Proposition 1 in [3], but we need a few more details. We start by revisiting the sparsification lemma.

▶ **Lemma 9** (Sparsification Lemma, [30]). *For $\varepsilon > 0$ and $k \in \mathbb{N}$, there exists a constant $C = C(\varepsilon, k)$, such that any k-SAT formula F with $\tilde{n}$ variables can be expressed as $F = \bigvee_{i=1}^{l} F_i$, where $l = O(2^{\varepsilon\tilde{n}})$ and each F_i is a k-SAT formula with at most $C\tilde{n}$ clauses. This disjunction can be computed in time $O(2^{\varepsilon\tilde{n}}\operatorname{poly}(\tilde{n}))$.*

We set $C(\varepsilon)$ to the smallest $C(\varepsilon, k)$, over all k such that k-SAT cannot be solved faster than in $O^*(2^{(1-\varepsilon)\tilde{n}})$ time[2]; formally, $C(\varepsilon) = \min\{C(\varepsilon', k) : \varepsilon' < \varepsilon$ and $k \in \mathbb{N}$ with k-SAT $\notin O^*(2^{(1-\varepsilon')\tilde{n}})\}$. Note that $C(\varepsilon)$ is well-defined if we assume that SETH is true (see also

[2] The $O^*(\cdot)$ notation hides $\operatorname{poly}(\tilde{n})$ factors.

Proposition 1 in [3]). Finally, for any $\varepsilon > 0$ and $t \in \mathbb{N}$, we set $K(\varepsilon, t) = C(\varepsilon) \cdot t/(1-\varepsilon)$, which gives the sensitivity of our reductions.

In our reductions, $t \in \mathbb{N}$ is the exponent of the allowed preprocessing time and $\varepsilon > 0$ denotes the improvement in the exponent of the running time over the 2^n-time algorithm. We note that $K(\varepsilon, t)$ gives a tradeoff: For small t (i.e., less preprocessing time), the lower bounds hold for smaller sensitivities; a smaller choice of ε yields larger sensitivities.

In the reductions we will write K to denote $K(\varepsilon, t)$ and c to denote $C(\varepsilon, k)$ whenever it is clear from the context.

The Reductions

Our reductions are conceptually similar to the ones in [3], but the graph instances we construct are based on a novel idea to minimize the size of the batch updates we need to perform. Here we describe the construction of the graphs we use in the reductions and refer to Appendix A.8 for full proofs.

We give two graphs, H_δ and D_δ, for $\delta \in (0, 1)$. For the construction, let F be a SAT formula over a set V of $\tilde{n}$ variables and $c \cdot \tilde{n}$ clauses. Let $U \subset V$ be a subset of $\delta\tilde{n}$ variables.

Construction of H_δ: For each partial assignment to the variables in U we introduce a node. The set of these nodes is denoted by $\bar{U}$. For each clause of F we introduce a node and denote the set of these nodes by C. We add an edge between a partial assignment $\bar{u} \in \bar{U}$ and a clause $c \in C$ if $\bar{u}$ does not satisfy c. Observe that H_δ has $O(2^{\delta\tilde{n}})$ vertices and $O^*(2^{\delta\tilde{n}})$ edges.

Construction of D_δ: We partition the set of clauses C into $K = c/\delta$ groups of size $\delta\tilde{n}$ each and denote these groups by $G_1, \ldots, G_K$. For all groups G_i, we introduce a vertex into the graph for each non-empty subset g of G_i. The edges to and from the nodes of D_δ will be introduced during reductions. Observe that for each group we introduce $O(2^{\delta\tilde{n}})$ vertices and D_δ has $O(K \cdot 2^{\delta\tilde{n}})$ vertices in total.

Our reductions have small sensitivity since we will only need to insert a single edge from H_δ to each group of clauses in D_δ. Hence, we only need to insert $K = O(1)$ edges in order to connect H_δ and D_δ at each stage in the reduction. However, we will need to argue how we can efficiently pick the correct sets in D_δ.

5 Diameter Upper Bound

In this section, we present deterministic algorithms, which can compute a $(1+\varepsilon)$-approximation for the eccentricity, the radius and the diameter of directed and undirected unweighted graphs after single edge deletions. All of these algorithms run in time truly subcubic time for graphs with a truly subquadratic number of edges.

Bernstein and Karger [10] give an algorithm for the related problem of all-pairs shortest paths in a directed weighted graph $G = (V, E)$ in case of single edge deletions. Their oracle data structure requires $\tilde{O}(mn)$ preprocessing time. Given a triplet $(u, v, e) \in V^2 \times E$, the oracle can output the distance from u to v in $G \setminus e$ in $O(1)$ time.

For the diameter problem with single edge deletions, note that only deletions of the edges in the shortest paths trees can have an effect on the diameter. Using this property, a trivial algorithm to compute the exact diameter after the deletion of a single edge works as follows: Build the oracle data structure of Bernstein and Karger [10]. For each vertex v, consider its shortest paths tree T_v. Delete each tree-edge once and query the distance from v to u in $G \setminus e$ for all vertices u in the subtree of the deleted tree-edge. By keeping track of the maximum distances, the diameter of G after a single edge deletion is computed exactly. As

there are $n-1$ edges in T_v, we spend $O(n^2)$ time for each vertex. Thus, the trivial algorithm requires $O(n^3)$ time.

In this section, we improve upon this result as follows.

▶ **Theorem 10.** *Let $G=(V,E)$ be a directed or undirected unweighted graph with n vertices and m edges, let $\varepsilon > 0$, and let D be the diameter of G. There exists a data structure that given a single edge $e \in E$ returns for $G \setminus e$ in constant time (1) the diameter, (2) the radius, and (3) the eccentricity of any vertex $v \in V$ within an approximation ratio of $1+\varepsilon$. It takes $\tilde{O}(n^{1.5}\sqrt{Dm/\varepsilon} + mn)$ preprocessing time to build this data structure.*

The rest of this section is devoted to the proof of the theorem. We give the proof of the theorem for directed graphs and point out the same proof also works for undirected graphs. We first describe how we can answer queries for the eccentricity of a fixed vertex $v \in V$ after a single edge deletion. After this, we explain how to extend this algorithm to solve the diameter and the radius problems after single edge deletions, and analyse the correctness and running time of the algorithm.

The data structure preprocesses the answers to all queries. Then queries can be answered via table lookup in $O(1)$ time.

Preliminaries

Let $G=(V,E)$ be a directed unweighted graph. For two vertices $u, u' \in V$ we denote the distance of u and u' in G by $d_G(u,u')$. For an edge $e \in E$ and vertices $u, u' \in V$, we denote the distance in the graph $G \setminus e$ by $d_{G\setminus e}(u,u')$. Given a tree T with root v and a tree-edge $e \in T$, we denote the subtree of T that is created when e is removed from T and that does *not* contain v by T_e. We let d_e be the height of T_e. A node u in T has *level* i, if $d_G(v,u) = i$.

Let $F \in \mathbb{N}$ be some suitably chosen parameter (see the last paragraph* of this section). Then given a tree T with root v, we call a tree-edge e *high*, if both of its endpoints have level less than F from v; we call all other edges *low*. We denote the set of all high edges by $T_<$, i.e., $T_< = \{e=(w,w') \in T : d_G(v,w) < F, d_G(v,w') < F\}$; the set of all low edges is given by $T_>$.

The Algorithm

Our data structure preprocesses the answers to all queries, and then queries can be answered via table lookup in $O(1)$ time. The preprocessing has three steps: First, in the initialization phase, we compute several subsets of vertices that are required in the next steps. Second, we compute the eccentricity of v after the deletion of a high edge exactly. We compute it exactly, since after the deletion of a tree-edge high up in the shortest path tree T_v of v, the nodes close to v in T_v might "fall down" a lot. This possibly affects all vertices in the corresponding subtrees and, hence, we need to be careful which changes occur after deleting a high edge. On the other side, the relative distance of nodes which are "far away" from v in G before any edge deletion cannot increase too much. Thus, we simply estimate their new distances in the third step. More precisely, in the third step we compute a $(1+\varepsilon)$-approximation of the eccentricity of v after the deletion of a low edge.

Step 1: Initialization. We build the data structure of Bernstein and Karger [10], in mn time which for each triplet (u,v,e) can answer queries of the form $d_{G\setminus e}(u,v)$ in $O(1)$ time.
We compute the shortest path tree $T = T_v$ of v in time $O(nm)$ and denote its depth by d_v. By traversing T bottom-up, we compute the height d_e of the subtree T_e for each

tree-edge e; this takes time $O(n)$. We further construct the sets $T_<$ and $T_>$ of high and low tree-edges, respectively.

Fix $\varepsilon > 0$. We construct a set $S_v \subset V$ as follows: First add v to S_v. Then add each $u \in V$ which has the following two properties: (1) u is at level $i\varepsilon F$ for some integer[3] $i > 0$ and (2) there exists a node u' in the subtree of u in T_v, such that u' has distance $\varepsilon F/2$ in T_v from u. Note that we can add the root, but every other node we add can be charged to the $\varepsilon F/2$ parent nodes that come before it. Thus, we can have at most $1 + \frac{2n}{\varepsilon F}$ nodes in S_v. Note that for every $z \in V$ there exists a $y \in S_v$, s.t. y is an ancestor of z in T_v and there exists a path from y to z in T_v of length at most εF.

Using a second bottom-up traversal of T, for each tree-edge $e \in T$, we compute the set $S_e = T_e \cap S_v$, i.e., the intersection of the vertices in T_e and those in S_v. This can be done in $O(n)$ time by, instead of storing S_e explicitly for each edge $e = (w, w')$, storing a reference to the set containing the closest children of w' which are in S_v; then S_e can be constructed in $O(|S_e|)$ time by recursively following the references.

For each non-tree-edge e, we store d_v as the value for the eccentricity of v when e is deleted.

Step 2: Handling high edges. For each level $j = 1, \ldots, F-1$, we proceed as follows. We consider each tree-edge $e = (w, w') \in T_<$ with $d(v, w) < d(v, w') = j$, there are at most n of these. We build a graph G_e containing all nodes of T_e together with a additional directed path P of length $d_e + 4$ with startpoint r. The nodes in P are new vertices added to G_e. Each edge on P has weight 1, except the single edge incident to r, which has weight $d_G(v, w') - 1$. Additionally to the path, the graph contains as edges: (1) all edges from E, which have both endpoints in T_e, and (2) for each $e = (z, z')$ which has its startpoint $z \notin T_e$ and its endpoint $z' \in T_e$, an edge (z'', z'), where z'' is the node on P with distance $d_G(v, z)$ from r.

Observe that G_e has the property that all distances after the deletion of e are maintained *exactly*: By construction, the shortest path from r to $u \in T_e$ in G_e has exactly length $d_{G\setminus e}(v, u)$ (we prove this formally in Lemma 11).

After building G_e, we compute its depth starting from node r and store this value for edge e.

Step 3: Handling low edges. For each tree-edge $e = (w, w') \in T_>$, we do the following: Let $S = S_e \cup \{w'\}$. As answer for a deleted edge e, we store $\max\{d_v, (1+\varepsilon)\max_{y \in S} d_{G\setminus e}(v, y)\}$. To determine $d_{G\setminus e}(v, y)$, we use the data structure of [10].

Extension to $(1+\varepsilon)$-approximate Diameter and Radius. We repeat the previously described procedure for all $v \in V$ (but we build the data structure for Bernstein and Karger only once). To compute the diameter, we keep track of the maximum value we encounter for each deleted edge e. To compute the radius, we keep track of the minimum value we encounter for each deleted edge e.

Correctness

Observe that it is enough to show correctness for a fixed $v \in V$. We first prove the correctness of the algorithm after removing high edges.

▶ **Lemma 11.** *After the deletion of a high edge $e \in T_<$, we compute the eccentricity of v exactly.*

[3] For readability we leave out the floors, however, we are considering the integer levels $\lfloor i\varepsilon F \rfloor$.

Proof. Consider any vertex $u \in T_e$ and consider the shortest path p from v to u in $G \setminus e$.

We can assume that p has exactly one edge (z, z'), s.t. $z \notin T_e$ and $z' \in T_e$: Assume that there is a path p' with two edges (x, x') and (y, y'), s.t. $x, y \notin T_e$, $x', y' \in T_e$ and y appears later on p' than x. Since $y \notin T_e$, there exists a path from v to y that does not use any vertex from T_e and that is of the same length as the subpath of p' from v to y in T, because T is a shortest-path tree with root v. Hence, we can choose a path to y without entering T_e.

Let z, z' be as before. Then $d_{G\setminus e}(v, u) = d_{G\setminus e}(v, z) + 1 + d_G(z', u)$. By construction of G_e, there exists a vertex on P in G_e with distance $d_{G\setminus e}(v, z)$ from r and which has an edge to z'. All paths which only traverse vertices from T_e are unaffected by the deletion of e. Hence, in G_e there exists a path of length $d_{G\setminus e}(u, v)$.

Also, there is no shorter path in G_e from r to u, because this would imply a shorter path in $G \setminus e$ by construction. ◀

Next we prove the correctness of the algorithm after the removal of low edges. Consider a tree-edge $e = (w, w') \in T_>$ with $F \geq d(v, w') > d(v, w)$. Let $S = S_e \cup \{w'\}$.

▶ **Lemma 12.** *For each node $z \in T_e$, there exists a vertex $y \in S$ s.t. $d_{G\setminus e}(y, z) \leq \varepsilon F$.*

Proof. Since $w' \in S$, the claim is true for all nodes $z \in T_e$ with $d(w', z) \leq \varepsilon F$. By construction of S_v, any (directed) tree path of length εF contains a node of S_v. For any node $z \in T_e$ with $d(w', z) > \varepsilon F$, there exists an ancestor u of z in T_e with $d(u, z) \leq \varepsilon F$. The path from u to z is a directed tree path and, thus, must contain a node in S_v. Thus, for each node in T_e there is a path of length at most εF from some node in S_v. ◀

▶ **Lemma 13.** *Consider two vertices $y, z \in T_e$ and assume there exists a path from y to z in $G \setminus e$. Then $d_{G\setminus e}(y, z) \leq X$ implies $d_{G\setminus e}(v, z) \leq d_{G\setminus e}(v, y) + X$.*

Proof. Concatenate the shortest paths from v to y in $G \setminus e$ and from y to z in $G \setminus E$, which both avoid e. This path cannot be shorter than the shortest path from v to z in $G \setminus e$. ◀

We define the maximum height achieved by the vertices of T_e in $G \setminus e$ by

$$n(v, e) = \max_{z \in T_e} d_{G\setminus e}(v, z).$$

Notice that the eccentricity of v in $G \setminus e$ is given by $\max\{d_v, n(v, e)\}$. Hence, by giving a $(1 + \varepsilon)$-approximation of $n(v, e)$, we obtain a $(1 + \varepsilon)$-approximation for the eccentricity of v in $G \setminus e$. In the remainder of this subsection, we show this guarantee on the approximation ratio of $n(v, e)$.

▶ **Lemma 14.** $n(v, e) \leq (1 + \varepsilon) \max_{y \in S} d_{G\setminus e}(v, y)$.

Proof. Let z' be any vertex in T_e such that $d_{G\setminus e}(v, z') = n(v, e)$. By Lemma 12 there exists a vertex $y' \in S$ with $d_{G\setminus e}(y', z') \leq \varepsilon F$. Then by Lemma 13,

$$\begin{aligned} n(v, e) &= d_{G\setminus e}(z', v) \\ &\leq d_{G\setminus e}(v, y') + \varepsilon F \\ &\leq \max_{y \in S} d_{G\setminus e}(v, y) + \varepsilon d_G(v, w') \\ &\leq (1 + \varepsilon) \max_{y \in S} d_{G\setminus e}(v, y), \end{aligned}$$

where in the second last step we used $F \leq d_G(v, w')$ and in the last step we used that $w' \in S$. ◀

▶ **Lemma 15.** $(1 + \varepsilon) \max_{y \in S} d_{G\setminus e}(v, y) \leq (1 + \varepsilon) n(v, e)$.

Proof. This follows from the definition of $n(v, e)$, since S is a subset of the vertices in T_e. ◀

Running Time Analysis.

Let us first consider the time spent on step 1, preprocessing. We build the data structure of Bernstein and Karger [10] in time $\tilde{O}(mn)$. For each node v, computing the shortest path tree of v takes time $\tilde{O}(m)$ and we spend time $O(n)$ computing the heights of the subtrees of T and computing the sets $T_<$, $T_>$, S_v. The sets S_e can as well be computed in $O(n)$ time by storing them only implicitly.

Now let us consider the time spent on step 2, the high edges. For the high edges e at level $j \leq F$, observe that the trees T_e are mutually disjoint. Hence, for a fixed level j, in time $\tilde{O}(m)$ we can compute the depths of *all* graphs G_e with e at level j. Since we have to do this for each level less than F, the total time for this step is $\tilde{O}(Fm)$.

Finally, let us consider the time spent on step 3, the low edges. For all low edges at level $j > F$, we query all nodes of S_v with height more than j. These are $O(\frac{n}{\varepsilon F})$ many such nodes. Thus, the total time we spend for all edges in T is $O(d_v \cdot \frac{n}{\varepsilon F})$.

To compute the diameter, we have to execute the above steps once for each $v \in V$, but we only need to build the data structure of Bernstein and Karger once. Hence, the total time is $\tilde{O}(mn + Fmn + nd_v \cdot \frac{n}{\varepsilon F})$. Denote the diameter of G by D. Then setting $F = \sqrt{\frac{Dn}{\varepsilon m}}$ yields a total running time of $\tilde{O}(n^{1.5}\sqrt{Dm/\varepsilon} + mn)$. Since $D \leq n$, this is $\tilde{O}(n^2\sqrt{m/\varepsilon})$.

References

1. Amir Abboud and Søren Dahlgaard. Popular conjectures as a barrier for dynamic planar graph algorithms. In *FOCS*, 2016.
2. Amir Abboud, Fabrizio Grandoni, and Virginia Vassilevska Williams. Subcubic equivalences between graph centrality problems, APSP and diameter. In *SODA*, pages 1681–1697, 2015.
3. Amir Abboud and Virginia Vassilevska Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *FOCS*, pages 434–443. IEEE, 2014.
4. Amir Abboud, Virginia Vassilevska Williams, and Huacheng Yu. Matching triangles and basing hardness on an extremely popular conjecture. In *STOC*, pages 41–50, 2015.
5. V. L. Arlazarov, E. A. Dinic, M. A. Kronrod, and I. A. Faradzev. On economical construction of the transitive closure of an oriented graph. *Soviet Math. Dokl.*, 11:1209–1210, 1970.
6. Surender Baswana, Keerti Choudhary, and Liam Roditty. Fault tolerant reachability for directed graphs. In *DISC*, pages 528–543, 2015.
7. Surender Baswana, Keerti Choudhary, and Liam Roditty. Fault tolerant subgraph for single source reachability: Generic and optimal. In *STOC*, pages 509–518, 2016.
8. Surender Baswana and Neelesh Khanna. Approximate shortest paths avoiding a failed vertex: Near optimal data structures for undirected unweighted graphs. *Algorithmica*, 66(1):18–50, 2013.
9. Surender Baswana, Utkarsh Lath, and Anuradha S. Mehta. Single source distance oracle for planar digraphs avoiding a failed node or link. In *SODA*, pages 223–232, 2012.
10. Aaron Bernstein and David Karger. A nearly optimal oracle for avoiding failed vertices and edges. In *STOC*, pages 101–110, 2009.
11. Davide Bilò, Fabrizio Grandoni, Luciano Gualà, Stefano Leucci, and Guido Proietti. Improved purely additive fault-tolerant spanners. In *ESA*, 2015.
12. Davide Bilò, Luciano Gualà, Stefano Leucci, and Guido Proietti. Fault-tolerant approximate shortest-path trees. In *ESA*, 2014.
13. Davide Bilo, Luciano Guala, Stefano Leucci, and Guido Proietti. Compact and fast sensitivity oracles for single-source distances. In *ESA*, 2016.

14 Davide Bilò, Luciano Gualà, Stefano Leucci, and Guido Proietti. Multiple-edge-fault-tolerant approximate shortest-path trees. In *STACS*, pages 18:1–18:14, 2016.

15 Gilad Braunschvig, Shiri Chechik, and David Peleg. Fault tolerant additive spanners. In *WG*, 2012.

16 Marco L. Carmosino, Jiawei Gao, Russell Impagliazzo, Ivan Mihajlin, Ramamohan Paturi, and Stefan Schneider. Nondeterministic extensions of the strong exponential time hypothesis and consequences for non-reducibility. In *ITCS*, pages 261–270, 2016.

17 T. M. Chan. More algorithms for all-pairs shortest paths in weighted graphs. In *STOC*, pages 590–598, 2007.

18 Shiri Chechik, Sarel Cohen, Amos Fiat, and Haim Kaplan. $1+\varepsilon$-approximate f-sensitive distance oracles. In *SODA*, 2017.

19 Shiri Chechik, Michael Langberg, David Peleg, and Liam Roditty. f-sensitivity distance oracles and routing schemes. *Algorithmica*, 63(4):861–882, 2012.

20 Keerti Choudhary. An optimal dual fault tolerant reachability oracle. In *ICALP*, pages 130:1–130:13, 2016.

21 Søren Dahlgaard. On the hardness of partially dynamic graph problems and connections to diameter. In *ICALP*, pages 48:1–48:14, 2016.

22 Ran Duan and Seth Pettie. Dual-failure distance and connectivity oracles. In *SODA*, pages 506–515, 2009.

23 Ran Duan and Seth Pettie. Connectivity oracles for failure prone graphs. In *STOC*, pages 465–474, 2010.

24 Ran Duan and Seth Pettie. Connectivity oracles for graphs subject to vertex failures. In *SODA*, 2017.

25 François Le Gall. Powers of tensors and fast matrix multiplication. In *ISSAC*, pages 296–303, 2014.

26 Fabrizio Grandoni and Virginia Vassilevska Williams. Improved distance sensitivity oracles via fast single-source replacement paths. In *FOCS*, pages 748–757, 2012.

27 Monika Henzinger, Sebastian Krinninger, Danupon Nanongkai, and Thatchaphol Saranurak. Unifying and strengthening hardness for dynamic problems via the online matrix-vector multiplication conjecture. In *STOC*, pages 21–30, 2015.

28 Monika Henzinger and Stefan Neumann. Incremental and fully dynamic subgraph connectivity for emergency planning. In *ESA*, 2016.

29 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

30 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

31 A. Itai and M. Rodeh. Finding a minimum circuit in a graph. *SIAM J. Computing*, 7(4):413–423, 1978.

32 Tsvi Kopelowitz, Seth Pettie, and Ely Porat. Higher lower bounds from the 3sum conjecture. In *SODA*, pages 1272–1287, 2016.

33 E. Nardelli, G. Proietti, and P. Widmayer. A faster computation of the most vital edge of a shortest path. *Information Processing Letters*, 79(2):81–85, 2001.

34 Merav Parter. Dual failure resilient BFS structure. In *PODC*, pages 481–490, 2015.

35 Merav Parter and David Peleg. Fault tolerant approximate bfs structures. In *SODA*, pages 1073–1092, 2014.

36 Mihai Patrascu. Towards polynomial lower bounds for dynamic problems. In *STOC*, pages 603–610, 2010.

37 Mihai Patrascu and Mikkel Thorup. Planning for fast connectivity updates. In *FOCS*, pages 263–271, 2007.

38 L. Roditty and U. Zwick. Replacement paths and k simple shortest paths in unweighted directed graphs. In *ICALP*, pages 249–260, 2005.
39 Liam Roditty and Uri Zwick. Replacement paths and k simple shortest paths in unweighted directed graphs. *ACM Trans. Algorithms*, 8(4):33, 2012.
40 Virginia Vassilevska Williams. Faster replacement paths. In *SODA*, pages 1337–1346, 2011.
41 Virginia Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd. In *STOC*, pages 887–898, 2012.
42 Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *FOCS*, pages 645–654, 2010.

A Appendix

A.1 Definitions of the Problems

We give definitions of the problems we consider in this paper in Table 1.

A.2 A Note on Terminology

The terminology used in the literature for dynamic data structures in the spirit of Section 1 is not consistent. The phrases which are used contain "fault-tolerant algorithms", "algorithms with sensitivity", "algorithms for emergency planning" and "algorithms for failure prone graphs".

In the community of spanners and computational graph theory, it is common to speak about "fault-tolerant subgraphs". In this area, this term is used consistently.

In the dynamic graph algorithms community, multiple phrases have been used to describe algorithms for the model proposed in Section 1. First, the field was introduced by [37] as algorithms for "emergency planning". Later, the terminologies "sensitivity" and "failure prone graphs" were used (e.g., [19, 18, 23, 24]). When the number of failures in the graph was fixed (e.g. 1 or 2), then often this was stated explicitly (without further mentioning sensitivity or failure prone graphs). However, it appears that in the dynamic graph algorithms community the phrase "sensitivity" is the most widely used one.

A.3 A Note on Adversaries

Some of the conditional lower bounds we obtain are for *randomized* algorithms. Previous literature [3, 27, 32] also gave conditional lower bounds for randomized dynamic algorithms; however, it was not discussed under which kind of adversary the obtained lower bounds hold. This depends on the conjecture from which the lower bound was obtained. We observe that in reductions from the static triangle problem, the only randomness is over the input distribution of the static problem. Hence, for lower bounds from the triangle conjecture, we can assume an oblivious adversary. Furthermore, we assume the OMv conjecture in its strongest possible form, i.e. for oblivious adversaries. (In [27] the authors did not explicitly state which kind of adversary they assume for their conjecture.) Thus, all conditional lower bounds we obtain for randomized algorithms hold for oblivious adversaries. Note that a lower bound which holds for oblivious adversaries must always hold for non-oblivious ones.

We would like to point out another subtlety of our lower bounds: In reductions from the triangle detection conjecture, the running time of the algorithm is assumed to be a random variable, but the algorithm must always answer correctly. However, in reductions from OMv the running time of the algorithm is determinstic, but the probability of obtaining a correct answer must be at least $2/3$.

Table 1 The problems we consider in this paper.

Problem		
Maintain	Update	Query
Reachability		
Directed graph	Edge insertions/deletions	Given two vertices u, v, can v be reached from u?
#SSR		
Directed graph and a fixed source vertex s.	Edge insertions/deletions	How many vertices can be reached from s?
Strong Connectivity (SC)		
Directed graph	Edge insertions/deletions	Is the graph strongly connected?
2 Strong Components (SC2)		
Directed graph	Edge insertions/deletions	Are there more than 2 SCCs?
2 vs k Strong Components (AppxSCC)		
Directed graph	Edge insertions/deletions	Is the number of SCCs 2 or more than k?
Maximum SCC Size (MaxSCC)		
Directed graph	Edge insertions/deletions	What is the size of the largest SCC?
Subgraph Connectivity		
Fixed undirected graph, with some vertices on and some off.	Turn on/off vertex	Given two vertices u, v, are u and v connected by a path only traversing vertices that are on?
α-approximate Shortest Paths		
Directed or undirected (possibly weighted) graph	Edge insertions/deletions	Given two vertices u, v, return an α-approximation of the length of the shortest path from u to v.
α-approximate Eccentricity		
Undirected graph	Edge insertions/deletions	Given a vertex u, return an α-approximation of the eccentricity of v.
α-approximate Radius		
Undirected graph	Edge insertions/deletions	Return an α-approximation of the radius of the graph.
α-approximate Diameter		
Undirected graph	Edge insertions/deletions	Return an α-approximation of the diameter of the graph.
Bipartite Perfect Matching (BPMatch)		
Undirected bipartite graph	Edge insertions/deletions	Does the graph have a perfect matching?
Bipartite Maximum Weight Matching (BWMatch)		
Undirected bipartite graph with integer edge weights	Edge insertions/deletions	Return the weight of the maximum weight perfect matching.

A.4 Our Lower Bounds

In Table 2 we summarize summarize our lower bounds for sensitivity data structures. Table 3 states our lower bounds for static oracle data structures.

A.5 Existing Sensitivity Data Structures

In Table 4 we summarize existing sensitivity data structures.

In the table, we also list algorithms for "fault-tolerant subgraphs" although they are not algorithms for the sensitivity setting in the classical sense. However, the fault-tolerant subgraphs are often much smaller than the input graphs and by traversing the fault-tolerant subgraph during queries, one can obtain better query times than by running the static algorithm on the original graph. Unfortunately, the construction time of these subgraphs is often very expensive, though still polynomial; the goal of these papers is to optimize the trade-offs between the size of the subgraphs and the approximation ratios achieved for the specific problem.

It is striking that (to the best of our knowledge) most of the existing algorithmic work was obtained for the case of *decremental* algorithms with a limited number of failures. While this is natural for the construction of fault-tolerant subgraphs, this is somewhat surprising from an algorithmic point of view. Our lower bounds might give an explanation of this phenomenon as they indicate that for many problems there is a natural bottleneck when it comes to the insertion of edges.

A.6 Triangle Detection Proofs

We provide full details of the reachability and shortest paths sensitivity results from Theorem 5.

Reachability

Let $G = (V, E)$ be an undirected unweighted graph for Triangle Detection. We create four copies of V denoted by V_1, V_2, V_3, V_4, and for $i = 1, 2, 3$, we add edges between nodes $u_i \in V_i$ and $v_{i+1} \in V_{i+1}$ if $(u, v) \in E$.

For a fixed source $s \in V$ and sink $t \in V$, Abboud and Williams [3] give the following reduction: For each vertex $v \in V$, they insert the edges (s, v_1) and (v_4, t), and query if there exists a path from s to t. They show that there exists a triangle in G iff one of the queries is answered positively. We observe that this reduction requires n batch updates of size 2 and n queries. Hence, it holds for sensitivity 2.

Now keep s fixed, but remove the sink t, and allow single-source reachability queries[4]. We perform a stage for each $v \in V$, in which we add the single edge (s, v_1) and query if there exists a path from s to v_4. By the same reasoning as before, there exists a triangle in G iff one of the queries returns true. The reduction requires n updates of size 1 and n queries. Thus, it has sensitivity 1.

Finally, we remove the source node s and ask all-pairs reachability queries[5]. We perform a stage for each $v \in V$, which queries if there exists a path from v_1 to v_4. There exists a triangle in G iff one if the queries returns true. This reduction has sensitivity 0, i.e., it uses

[4] Given $v \in V$, a query returns true iff there exists a path from s to v.
[5] Given two nodes $u, v \in V$, a query returns true iff there exists a path from u to v.

Table 2 The conditional lower bounds we obtained for non-zero sensitivity. Problems for which there exists a tight upper bound are marked bold. Regarding the sensitivities, the lower bounds hold for any data structure that supports *at least* the sensitivity given in the table; d is a parameter that can be picked arbitrarily, and $K(\varepsilon, t)$ is a constant depending on properties of SAT and the allowed preprocessing time (see Section 4). Lower bounds for constant sensitivities hold in particular for any dynamic algorithm which allows for *any* larger fixed constant sensitivity or sensitivity $\omega(1)$. For the query type we use the following abbreviations: *st* – fixed source and sink, *ss* – single source, *ap* – all pairs, *ST* – a fixed set of sources and a fixed set of sinks. The rest of the abbreviations are as follows: sh. paths means shortest paths, conn. means connectivity, SC means strongly connected components, SC2 means whether the number of strongly connected components is more than 2, Reach. means Reachability, BPMatch is bipartite matching, BWMatch is bipartite maximum weight matching, ecc. is eccentricity, dir. means directed, und. means undirected, w. means weighted, unw. means unweighted, Conj. means Conjecture.

Problem	**Inc/Dec**	**Query Type**	**Lower Bounds** $p(m,n)$	$u(m,n)$	$q(m,n)$	**Sens.**	**Conj.**	**Cite**
Reachability	Inc	*ss*	$\mathbf{n^{3-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	1	BMM	Theorem 5
Reach., SC, BPMatch		*st*	$\mathbf{n^{3-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	2		[3]
$(3/2-\varepsilon)$-sh. paths (und. unw.)	Inc	*ss*	$\mathbf{n^{3-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	1	BMM	Theorem 5
$(7/5-\varepsilon)$-sh. paths (und. unw.)		*st*	$\mathbf{n^{3-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	2	BMM	Theorem 5
#SSR	Inc	*ss*	$\mathbf{n^{t}}$	$\mathbf{m^{1-\varepsilon}}$	$\mathbf{m^{1-\varepsilon}}$	$K(\varepsilon,t)$	SETH	Lemma 18
reachability,	Inc	*ST*	n^t	$n^{1-\varepsilon}$	$n^{1-\varepsilon}$	$K(\varepsilon,t)$	SETH	Lemmas 20 and 19
$(4/3-\varepsilon)$-diameter for sparse graphs		- -	poly($\mathbf{n}$)	$\mathbf{n^{2-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	$\omega(\log n)$	SETH	[3]
SC2, AppxSCC, and MaxSCC	Inc	- -	poly($\mathbf{n}$)	$\mathbf{m^{1-\varepsilon}}$	$\mathbf{m^{1-\varepsilon}}$	$\omega(\log n)$	SETH	[3]
subgraph conn. ($\Longrightarrow$ reachability,	Inc	*st*	poly($\mathbf{n}$)	poly($\mathbf{d}$)	$\mathbf{d^{1-\varepsilon}}$	d	OMv	Theorem 21
BPMatch, SC)			$n^{2-\varepsilon}$	$n^{1-\varepsilon}$	$d^{1-\varepsilon}$	d	3SUM	[32]
$(2-\varepsilon)$-sh. paths	Inc	*ss*	poly(n)	poly(d)	$d^{1-\varepsilon}$	d	OMv	Theorem 21
$(5/3-\varepsilon)$-sh. paths ($\Longrightarrow$ BWMatch)	Inc	*st*	poly(n)	poly(d)	$d^{1-\varepsilon}$	d	OMv	Theorem 21
diameter $(4/3-\varepsilon)$, und. unw.	Dec	-	$\mathbf{n^{3-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	1	BMM	Theorem 5
dir. & und. w.	Dec	-	$\mathbf{n^{3-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	1	APSP	Section 3
$(4/3-\varepsilon)$-ecc.	Dec	-	$\mathbf{n^{3-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	1	BMM	Theorem 5
weighted-ecc.	Dec	-	$\mathbf{n^{3-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	1	APSP	Lemma 17
shortest paths								
dir. w.	Dec	*st*	$\mathbf{n^{3-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	1	APSP	[42]
und. w.	Dec	*st*	$\mathbf{n^{3-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	2	APSP	Section 3
reachability	Dec	*st*	$\mathbf{n^{3-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	$\Omega(\log n)$	BMM	[3]
($\Longrightarrow$ SC, BPMatch)							BMM	[3]
shortest paths (undir. unw.)	Dec	*st*	$\mathbf{n^{3-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	$\mathbf{n^{2-\varepsilon}}$	$\Omega(\log n)$	BMM	Theorem 5
subgraph conn. ($\Longrightarrow$ reachability,	Dec	*st*	poly($\mathbf{n}$)	poly($\mathbf{d}$)	$\mathbf{d^{1-\varepsilon}}$	d	OMv	Theorem 21
BPMatch, SC)			$n^{2-\varepsilon}$	$n^{1-\varepsilon}$	$d^{1-\varepsilon}$	d	3SUM	[32]
$(2-\varepsilon)$-sh. paths	Dec	*ss*	poly(n)	poly(d)	$d^{1-\varepsilon}$	d	OMv	Theorem 21
$(5/3-\varepsilon)$-sh. paths ($\Longrightarrow$ BWMatch)	Dec	*st*	poly(n)	poly(d)	$d^{1-\varepsilon}$	d	OMv	Theorem 21

Table 3 The conditional lower bounds we obtained for static oracle data structures, i.e., data structures with zero sensitivity. Problems for which there exists a tight upper bound are marked bold. The query type "ap" denotes all pairs queries, Repl. means replacement, the rest of the abbreviations are as in Table 1.

Problem	Inc/Dec/ Static	Query Type	Lower Bounds $p(m,n)$	$u(m,n)$	$q(m,n)$	Conj.	Cite
Reach.	static	*ap*	$\mathbf{n^{3-\varepsilon}}$	-	$\mathbf{n^{2-\varepsilon}}$	BMM	Theorem 5
$(5/3-\varepsilon)$-sh. paths	static	*ap*	$\mathbf{n^{3-\varepsilon}}$	-	$\mathbf{n^{2-\varepsilon}}$	BMM	Theorem 5
Repl. paths (1 edge fault) (dir. w.)	static	*st*	$\mathbf{n^{3-\varepsilon}}$	-	$\mathbf{n^{2-\varepsilon}}$	APSP	Section A.7

no updates, and n queries. Hence, we have derived a very simple conditional lower bound for static reachability oracles.

These reductions prove the first three results of Theorem 5.

Shortest Paths

The above reduction for *st*-reachability can be easily altered to work for $(7/5-\varepsilon)$-approximate *st*-shortest paths in *undirected* unweighted graphs (for any $\varepsilon > 0$) with the same running time lower bounds: Just observe that the graphs in the reduction are bipartite. Thus, either there is a path from s to t of length 5 and there is a triangle in the original graph, or the shortest path between s and t has length at least 7. Thus distinguishing between length 7 and 5 solves the triangle problem.

To obtain a lower bound for $(3/2-\varepsilon)$-approximate *ss*-shortest paths, we take the construction for *ss*-reachability and again observe that the graph is bipartite so that if there is no path of length 4 between s and a node $v \in V$, then the shortest path between them must have length at least 6.

With the same bipartiteness observation, we obtain a conditional lower bound for $(5/3-\varepsilon)$-approximate static *ap*-shortest paths. In a stage for node $v \in V$, we query the shortest path from v_1 to v_4. The query returns 3 if there exists a triangle in the original graph containing the vertex v and ≥ 5 otherwise. Thus, distinguishing between 3 and ≥ 5 suffices to solve the problem.

Finally, let us discuss how we obtain a lower bound for incremental shortest paths with sensitivity 1 in directed graphs. Vassilevska Williams and Williams [42] reduce BMM to the replacement paths problem in directed unweighted graphs: given a directed graph G on m edges and n nodes and two nodes s and t, compute for every $e \in E$, the distance between s and t in $G \setminus \{e\}$. [42] shows that if a combinatorial algorithm can solve the latter problem in $O(mn^{1/2-\varepsilon})$ time for any $\varepsilon > 0$ (for any choice of m as a function of n), then BMM has a truly subcubic combinatorial algorithm. This showed that the $O(m\sqrt{n})$ time algorithm of Roditty and Zwick [39] is tight.

Here we observe that the [42] reduction immediately implies a 1-sensitivity oracle lower bound, as any 1-sensitivity oracle for *st* distances must be able to answer the replacement paths problem by querying the less than n nodes on the shortest s-t path: either the preprocessing time is at least $mn^{0.5-o(1)}$ or the query time is at least $m/n^{0.5+o(1)}$. For dense graphs this gives a lower bound of either $n^{2.5-o(1)}$ preprocessing or $n^{1.5-o(1)}$ query time. The lower bound is again tight via Roditty and Zwick's algorithm [39] for replacement paths.

Table 4 Upper Bounds. We omit polylog factors in the stated running times and spaces usages. We use the following abbreviations: "ap" means "all pairs", "ss" means "single source", "st" denotes problems with a fixed source and a fixed sink. Oracles combine update and query into a single operation. For algorithms with an additive approximation guarantee, we included the optimal result $\mathcal{O}$; all other approximation algorithms achieve multiplicative approximation guarantees. See table 5 for APSP upper bounds.

Problem	**Approx.**	**Upper Bounds** Space	$p(m,n)$	$u(m,n)$	$q(m,n)$	**Sensitivity**	**Ref.**	**Remark**
Dec ap-Connectivity		n	poly(n)	d	1	d	[37]	
Dec ap-SubgraphConn		$d^{1-2/c}mn^{1/c}$	$d^{1-2/c}mn^{1/c}$	d^{4+2c}	d	d	[23]	Any $c \in \mathbb{N}$ can be picked; space simplified.
		dm	mn	d^3	d	d	[24]	Deterministic.
		m	mn	d^2	d	d	[24]	Randomized.
Inc ap-SubgraphConn		n^2	n^3	d^2	d	d	[28]	
Fully Dynamic ap-SubgraphConn		n^2m	n^3m	d^4	d^2	d	[28]	Uses [24] as a blackbox.
Dec ss-Reachability		n	m	1	n	1	[6]	Subgraph.
(implies SCC, dominator tree)		n	n	1		1	[9]	Oracle. Planar graph.
		n	poly(n)	1		2	[20]	Allows for vertex failures.
		$2^d n$	$2^d mn$	1	$2^d n$	d	[7]	Subgraph. Reasonable for $d = o(\log(m/n))$.
Dec ss-SP								
undirected unweighted	3	n	m	1		1	[8]	Oracle.
	$1+\varepsilon$	n/ε^3		1		1	[8]	Oracle.
	exact	$n^{5/3}$	poly(n)	1	$n^{5/3}$	2	[34]	More robust BFS tree.
directed unweighted	exact	n^2	n^ω	1	1	1	[26]	Algorithm and oracle.
undirected weighted	$1+\varepsilon$	$m+n/\varepsilon$	mn	$1/\varepsilon$		1	[13]	Oracle.
	2	m	mn	1		1	[13]	Oracle.
	$2\mathcal{O}+1$	dn	dm	d^2		d	[14]	Oracle.

Table 5 Upper Bounds for APSP. We omit polylog factors in the stated running times and spaces usages. For algorithms with an additive approximation guarantee, we included the optimal result $\mathcal{O}$; all other approximation algorithms achieve multiplicative approximation guarantees.

Problem	**Approx.**	**Upper Bounds** Space	$p(m,n)$	$u(m,n)$	$q(m,n)$	**Sensitivity**	**Ref.**	**Remark**
Dec APSP								
unweighted undirected	$\mathcal{O}+2$	$n^{5/3}$	poly(n)	1	$n^{5/3}$	1	[11]	Additive spanner.
	$\mathcal{O}+4$	$n^{3/2}$	poly(n)	1	$n^{3/2}$	1	[11]	Additive spanner.
	$\mathcal{O}+10$	$n^{7/5}$	poly(n)	1	$n^{7/5}$	1	[11]	Additive spanner.
	$\mathcal{O}+14$	$n^{4/3}$	poly(n)	1	$n^{4/3}$	1	[11]	Additive spanner.
	$(2k-1)(1+\varepsilon)$	$kn^{1+(k\varepsilon^4)^{-1}}$		k		1	[8]	Oracle. Any $k>1$ and $\varepsilon>0$.
	3	n	poly(n)	1	n	1	[35]	Spanner.
	$3(d+1)\mathcal{O}+(d+1)\log n$	dn	poly(n)	1	dn	d	[35]	Spanner.
non-negative weights, undirected	$1+\varepsilon$	n/ε^2	poly(n)	1	n/ε^2	1	[12]	Spanner. Vertex and edge deletions.
weighted, undirected	$(8k+2)(d+1)$	$dkn^{1+1/k}$	poly(n)	d		d	[19]	Oracle. Any $k\in\mathbb{N}$.
	$1+\varepsilon$	$dn^2(\log n/\varepsilon)^d$	$dn^5\log(n/\varepsilon)^d$	d^5		d	[18]	Oracle.
weighted, directed	exact		$Mn^{2.88}$	$n^{0.7}$		1	[26]	Oracle. Weights: $\{-M,\ldots,M\}$. Simplified running times.
	exact	n^2	mn	1		1	[10]	Oracle.
	exact	n^2	poly(n)	1		2	[22]	Oracle.
	$\mathcal{O}+d$	$dn^{4/3}$	poly(n)	1	$n^{4/3}$	d	[15]	Additive spanner.

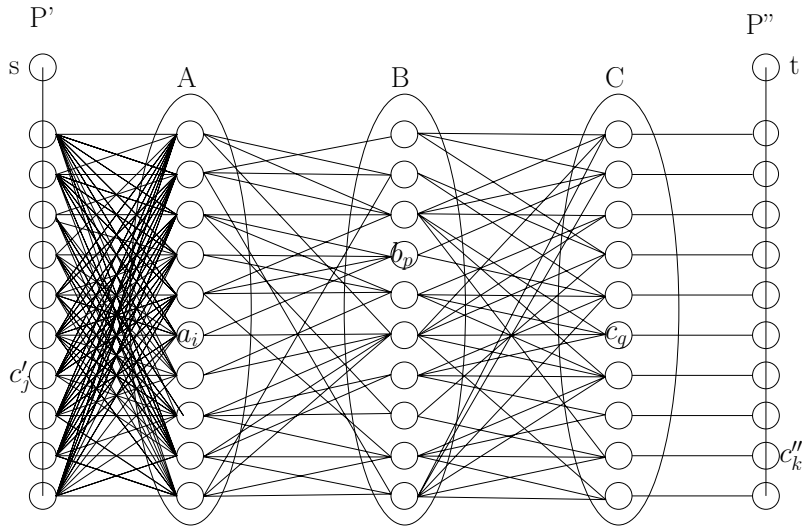


Figure 1 The graph G.

A.7 All Pairs Shortest Paths Proofs

We prove the statements of Theorem 7 in Lemma 16 and Lemma 17.

▶ **Lemma 16.** *Assuming the APSP conjecture, decremental st-shortest paths in undirected weighted graphs with sensitivity 2 cannot be solved with preprocessing time $O(n^{3-\varepsilon})$, and update and query times $O(n^{2-\varepsilon})$ for any $\varepsilon > 0$.*

Proof. We use a reduction similar to the reduction from APSP to RP from [42], but to deal with the undirected edges we add more weights and we add additional nodes to the graph. As in [42], we start by taking an instance of APSP and turning it into a tripartite graph for the negative triangle detection problem[6]; denote resulting graph H'.

If H' has no negative edge weights, we are done (there are no negative triangles). If H' has negative edge weights, let $M = \min\{w(e)|e \in E_{H'}\}$ and add $-M+1$ to all edges (thus making all edges have positive weights). Now we want to detect if there is a triangle with (positive) weight less than $-3M+3$. Denote the new graph by H and denote the three tripartite groups A, B and C; each set A, B and C has n nodes.

We construct a graph G in which n shortest paths queries with two edge deletions determine if there exist any triangles with weight less than $-3M+3$ in H. Let $W = 4\max\{w(e)|e \in E_H\}$ and observe that W is larger than the maximum possible difference in the weight of two triangles. We will use this weight to enforce that we must take certain paths.

We add two vertices s and t to G. We add a path P' of length n to s, where each edge on the path has weight 0; the first node after s on P' is denoted c'_1, the next node on P' is denoted c'_2, and the i'th node on P' is denoted c'_i. Next, we add a path P'' of length n to t, where each edge on the path has weight 0; the first node after t on P'' is denoted c''_n, second node on P'' is denoted c''_{n-1} and the i^{th} node away from t on P'' is denoted c''_{n-i+1}. We add the nodes in A, B and C from H to G and keep all edges from $A \times B$ and from $B \times C$,

[6] In the negative triangle detection problem we are given an edge-weighted graph $G = (V, E)$ with possibly negative edge-weights from $\mathbb{Z}$, and we must determine if G contains a triangle consisting of vertices u, v, x such that $w(u, v) + w(v, x) + w(x, u) < 0$.

however, we delete all edges from $A \times C$. We increase the weight of all edges from A to B and of all edges from B to C by $6nW$. We add edges between all nodes in A and all nodes on the path P'; specifically, for all $a_i \in A$ and for all $j \in [1, n]$, we add an edge (a_i, c'_j) of weight $(7n - j)W + w((a_i, c_j))$. We further add edges from C to the path P''; specifically, we add an edge from $c_i \in C$ to c''_i of weight $(6n + i)W$. The resulting graph is given in Figure 1.

Note that all edges in the graph G either have weight 0 or their weight is from the range $[6nW, 7nW + W/4]$. All edges of weight 0 are on the paths P' and P''; all paths from s to t must contain at least one edge from P' to A, one from A to B, one from B to C and finally one edge from C to P''. Each of the non-path-edges has weight from the range $[6nW, 7nW + W/4]$ and we must take at least four of them in total. If we backtrack (and go from A back to P' or from B back to A, etc) then we must take at least six non-zero edges; hence, it is never optimal to backtrack since $(7nW + W/4)4 < 6\dot{6}nW$.

We explain which n queries answer the negative triangle question in H. For each $i = 1, \ldots, n$, we delete the edge (c'_i, c'_{i+1}) from path P' and the edge (c''_i, c''_{i-1}) from path P'', and we query the shortest path from s to t. Note that with these edges deleted to take only four "heavy" edges, one must leave from a c'_j where $j \leq i$ and enter a c''_k where $k \geq i$. The length from s to c'_j and from c''_k to t is zero. So a shortest path from s to t has length

$$(7n - j)W + w(a_p, c_j) + w(a_p, b_q) + 6nW + w(b_q, c_k) + 6nW + (6n + k)W.$$

Note that because W is large and we want to minimize the length of the shortest path, we want to maximize j and minimize k. Due to the deleted edges, the maximum plausible value of j is i and the minimum plausible value of k is i. In that case, the path length is

$$(7n - i)W + w(a_p, c_i) + w(a_p, b_q) + 6nW + w(b_q, c_i) + 6nW + (6n + i)W.$$

This simplifies to $25nW + w(a_p, c_i) + w(a_p, b_q) + w(b_q, c_i)$. If the length of the shortest st-path is less than $25nW - 3M + 3$, then there exists a negative triangle in the graph H' containing c_i. Otherwise, there is no such triangle. ◀

▶ **Lemma 17.** *Assuming the APSP conjecture, decremental diameter in undirected weighted graphs with sensitivity 1 cannot be solved with preprocessing time $O(n^{3-\varepsilon})$, and update and query times $O(n^{2-\varepsilon})$ for any $\varepsilon > 0$.*

Proof. As in [42] we start by taking an instance of APSP and turning it into a weighted tripartite graph for the negative triangle detection problem; denote the resulting graph for the negative triangle detection problem H'.

Let M be a positive integer such that all edges in H' have weights between $-M$ and M. We increase all edge weights in H by $5M$ (thus making all edges have weight at least $4M$). We denote this new graph H and call the three tripartite groups X, Y and Z. Note that X, Y and Z each have n nodes. Now we want to detect if there is a triangle with weight less than $15M$ in H.

We construct the graph G depicted in Figure 2 as follows. We create sets V_1 and V_4 containing copies of the nodes in X, the set V_2 containing copies of the nodes from Y, and the set V_3 containing copies of the nodes from Z. We additionally create two groups of vertices, A and B, containing copies of X. Finally, we add two vertices c and d to G. For convenience, we denote the different copies of $x_i \in X$ as follows: The copy in V_1 as x_i^1, the copy in V_4 as x_i^4, the copy in A as x_i^A, and the copy in B as x_i^B.

We introduce edges between V_1 and V_2 if the corresponding nodes in X and Y have an edge; the edges between V_2 and V_3 are determined by the edges between Y and Z; the edges between V_3 and V_4 are determined by the corresponding edges between nodes of X and Z.

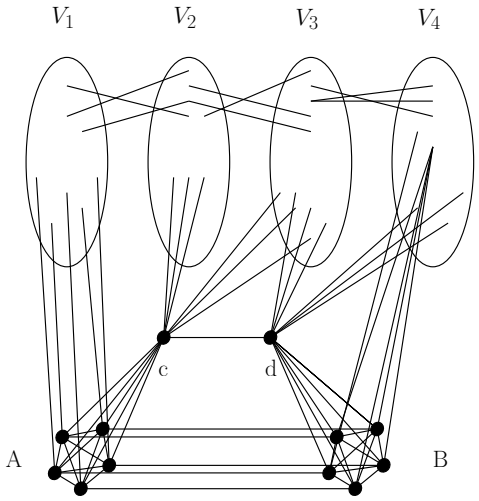


Figure 2 The graph G'.

All edges we added to the graph have the same weight as their corresponding copies in H. Note that there are no edges between V_1 and V_3, nor between V_1 and V_4, nor between V_2 and V_4.

For all $i = 1, \ldots, n$, we add edges of weight $4M$ between x_i^1 and x_i^A, and between x_i^A and x_i^B. Additionally, for all $i = 1, \ldots, n$ and all $j = 1, \ldots, n$, we add an edge between x_i^B and x_j^4 of weight $4M$. For all $v \in V_2 \cup V_3 \cup A$, we add an edge of weight $4M$ between c and v, and for all $v \in V_3 \cup V_4 \cup B$, we add an edge of weight $4M$ between d and v. We add edges with weight $4M$ to connect all vertices in A into a clique, and we do the same for B. Finally, we add an edge of weight $4M$ between c and d.

Note that all edges in this graph have weights between $4M$ and $6M$. Further note that in G all pairs of nodes have a path of length at most $12M$ between them.

Our n queries are picked as follows: For all $i = 1, \ldots, n$, we delete the edge between x_i^B and x_i^4 and query the diameter. Observe that the only path lengths that could become greater than $12M$ are between the nodes in V_1 and the node x_i^4. However, for all x_j^1 where $j \neq i$, the path $x_j^1 \to x_j^A \to x_j^B \to x_i^4$ still has all of its edges, and there exists a path of length $12M$ for these vertices.

Thus the only pair of vertices for the path length might increase is x_i^1 to x_i^4. One must use at most three edges to obatain a path of length at most $15M$ (since any path with four hops has length at least $16M$ because every edge in G has weight at least $4M$). Hence, any path of length less than $15M$ between x_i^1 and x_i^4 must go from x_i^1 to V_2 to V_3 to x_i^4. Thus, if a path of length less than $15M$ exists between x_i^1 and x_i^4 after the edge deletion between x_i^B and x_i^4, then there is a negative triangle in H' containing node x_i.

Thus, we can detect if any negative triangle exists in the tripartite graph H' by asking these n queries, determining for all $x_i \in X$ if a negative triangle containing x_i exists. ◀

Note that if for all $i = 1, \ldots, n$ we delete edge (x_i^B, x_i^4), then the eccentricity of x_i^1 is less than $15M$ if and only if the diameter of the graph is less than $15M$. Thus, this serves as lower bound for eccentricity as well.

A.8 SETH Proofs

▶ **Lemma 18.** *Let $\varepsilon > 0$, $t \in \mathbb{N}$. SETH implies that there exists no algorithm for incremental #SSR with sensitivity $K(\varepsilon, t)$, which has preprocessing time $O(n^t)$, update time $u(n)$ and query time $q(n)$, such that* $\max\{u(n), q(n)\} = O(n^{1-\varepsilon})$.

Proof. Set $\delta = (1-\varepsilon)/t$. Assume that the k-CNFSAT formula F has $c \cdot \tilde{n}$ clauses for some constant c. Partition the clauses into $K = c/\delta$ groups of size $\delta\tilde{n}$ and denote these groups by $G_1, \ldots, G_K$. Further let U be a subset of the $\tilde{n}$ variables of F of size $\delta\tilde{n}$, and let $\bar{U}$ denote the set of all partial assignments to the variables in the U.

We construct a graph G as the union of D_δ and H_δ. It consists of $\bar{U}$ and the set C of clauses. We direct the edges in H_δ from C to $\bar{U}$, and add a directed edge from a node $d \in D_\delta$ to $c \in C$ iff $c \in d$. We further add a single node s to the graph. The resulting graph has $n = O(2^{\delta\tilde{n}})$ nodes and $O(2^{\delta\tilde{n}}\tilde{n})$ edges.

We proceed in stages with one stage for each partial assignment to the variables in $V \setminus U$. At a stage for a partial assignment ϕ to the variables in $V \setminus U$, we proceed as follows: For each group $G_i \subset C$, we add an edge from s to the largest non-empty subset d_i of G_i which only contains clauses that are not satisfied by ϕ, i.e., to the set $d_i = \{c \in G_i : \phi \not\models c\}$; if d_i is empty, then we do not introduce an edge. Let D' be the set of nodes d_i that received an edge from s and let $d(s) = |D'|$. Note that $d(s) \leq K$, since we introduce at most one edge for each of the K groups. Further, let B denote the number of clauses in C reachable from the sets $d_i \in D'$, i.e., $B = \sum_{i=1}^{K} |d_i|$. We query if the number of nodes reachable from s is less than $d(s) + B + 2^{\delta\tilde{n}}$. If the answer to the query is true, then we return that F is satisfiable, otherwise, we proceed to the next partial assignment to the variables in $V \setminus U$.

We prove the correctness of the reduction: Assume that F is satisfiable. Then there exist partial assignments ϕ and ϕ' to the variables in $V \setminus U$ and U, such that $\phi \cdot \phi'$ satisfies F. Hence, for each subset of clauses $d \subset C$ we have that each clause $c \in d$ is satisfied by ϕ or by ϕ'. Thus, the node $u \in \bar{U}$ corresponding to ϕ' cannot be reachable from s and there must be less than $d(s) + B + 2^{\delta\tilde{n}}$ nodes reachable from s. Now assume that at a stage for the partial assignment ϕ the result to the query is true, i.e., less than $d(s) + B + 2^{\delta\tilde{n}}$ nodes are reachable from s; namely $d(s)$ at distance 1, B at distance 2, and less than $2^{\delta\tilde{n}}$ at distance 3. In this case, there must be a node $u \in \bar{U}$ which is not reachable from s: In D_δ there are exactly $d(s)$ nodes reachable and in C there are exactly B nodes reachable by construction of the graph and definition of $d(s)$ and B. This implies that for the partial assignment ϕ' corresponding to u, each clause $c \in C$ must be satisfied by ϕ or ϕ'. Hence, F is satisfiable.

Note that determining the sets $d_i \in D'$ can be done in time $O(\delta\tilde{n}^2)$ per group G_i as for each clause we can check in time $O(\tilde{n})$ whether it is satisfied by ϕ. Thus the set D' and the value B can be computed in total time $O(cn)$ and the total time for all stages is $O(2^{1-\delta\tilde{n}}(\tilde{n}^2 + Ku(n) + Kq(n))$. If both $u(n)$ and $q(n)$ are $O(n^{1-\varepsilon}) = O(2^{\delta\tilde{n}(1-\varepsilon)})$, then SAT can be solved in time $O^*(2^{\tilde{n}(1-\varepsilon\delta)})$. ◀

▶ **Lemma 19.** *Let $\varepsilon > 0$, $t \in \mathbb{N}$. SETH implies that there exists no $(4/3-\varepsilon)$-approximation algorithm for incremental diameter with sensitivity $K(\varepsilon, t)$, which has preprocessing time $O(n^t)$, update time $u(n)$ and query time $q(n)$, such that* $\max\{u(n), q(n)\} = O(n^{1-\varepsilon})$.

Proof. Set $\delta = (1-\varepsilon)/t$. During the proof we will assume that the k-CNFSAT formula F has $c \cdot \tilde{n}$ clauses for some constant c. We partition these clauses into $K = c/\delta$ groups of size $\delta\tilde{n}$ and denote these groups by $G_1, \ldots, G_K$.

We construct a graph as follows: For $U \subset V$ of size $\delta\tilde{n}$ we create the graph H_δ with the set of all partial assignments to the variables in U denoted by $\bar{U}$ and the set of clauses C.

We also add the graph D_δ containing all the subsets of the groups G_i. Furthermore, we introduce four additional nodes x, y, z and t. Observe that the graph has $O(2^{\delta\tilde{n}})$ vertices.

We add an edge from x to each node in $\bar{U}$. We connect y to all nodes in C and to all nodes in D_δ. We add further edges between a clause c and a set $g \in D_\delta$ if $c \in g$. We also add the following edges to E: $\{x, y\}$, $\{y, z\}$ and $\{z, t\}$. Hence, we have $O(2^{\delta\tilde{n}}\tilde{n})$ edges in total.

If during the construction of the graph we encounter that a clause is satisfied by all partial assignments in $\bar{U}$, then we remove this clause. Also, if there exists a partial assignment from $\bar{U}$ which satisfies all clauses, we return that the formula is satisfiable. Thus, we can assume that each node in $\bar{U}$ must have an edge to a node in C and vice versa.

We proceed in stages with one stage for each partial assignment to the variables in $V \setminus U$. Denote the partial assignment of the current stage by ϕ. For each G_i, we add an edge between t and the subset of G_i that contains all clauses of G_i which are not satisfied by ϕ, i.e., we add an edge between t and $\{c \in G_i : \phi \nvDash c\}$ for all $i = 1, \ldots, K$. Hence, in each stage we have $O(K) = O(1)$ updates. We query the diameter of the resulting graph. The diameter is 3, if the formula F is not satisfiable, and it is 4, otherwise. After that we remove the edges that were added in the update.

We prove the correctness of our construction: Observe that via x and y all nodes from $\bar{U} \cup C \cup G$ have a distance of at most 3. From z we can reach all vertices of G and C via y within two steps and all nodes of $\bar{U}$ within three steps via y and x. From t we can reach all nodes of $\{x, y, z\} \cup C \cup G$ within three steps using the path $t \to z \to y \to v$, where $v \in C \cup D_\delta \cup \{x\}$. Hence, all nodes in $\{x, y, z, t\} \cup C \cup G$ have a maximum distance of 3. From $\bar{u} \in \bar{U}$ we can reach t in four steps with the path $\bar{u} \to x \to y \to z \to t$.

Assume that for $\bar{u}$ there exists a path $\bar{u} \to c \to g \to t$, then by construction $\bar{u} \nvDash c$ and $\phi \nvDash c$, since $c \in g$. Hence, F is not satisfied by $\bar{u} \cdot \phi$. Thus, if the diameter is 3, then F is not satisfiable. On the other hand, if F is not satisfiable, then for each pair of partial assignments $\bar{u}$ and ϕ, there must be a clause c which both partial assignments do not satisfy. Hence, there must be a path of the form $\bar{u} \to c \to g \to \phi$ for some g with $c \in g$ and g has an edge to t. Thus, if F is not satisfiable, then the graph has diameter 3.

The sets d_i can be computed as in the previous proof. ◀

▶ **Lemma 20.** *Let $\varepsilon > 0$, $t \in \mathbb{N}$. SETH implies that there exists no algorithm for incremental ST-Reach with sensitivity $K(\varepsilon, t)$, which has preprocessing time $O(n^t)$, update time $u(n)$ and query time $q(n)$, such that $\max\{u(n), q(n)\} = O(n^{1-\varepsilon})$.*

Proof. We reuse the graph from the proof of Lemma 19. We update it by removing the vertices x, y, z. We further set $S = \bar{U}$ and $T = \{t\}$.

We proceed in stages with one stage for each partial assignment to the variables in $V \setminus U$. Denote the partial assignment of the current stage by ϕ. For each G_i, we add an edge between t and the subset of G_i that contains all clauses of G_i which are not satisfied by ϕ, i.e. we add an edge between t and $\{c \in G_i : \phi \nvDash c\}$ for all $i = 1, \ldots, K$. Hence, in each stage we have $O(K) = O(1)$ updates. We query for ST-Reachability. If the answer is true, then F is not satisfiable, otherwise, it is.

We prove the correctness of our construction: If F is not satisfiable, then for each pair of partial assignments $\bar{u}$ and ϕ, there must be a clause c which both partial assignments do not satisfy. Hence, there must be a path of the form $\bar{u} \to c \to g \to \phi$ for some g with $c \in g$ and g has an edge to t. Thus, if F is not satisfiable, then all nodes in S will be able to reach t. On the other hand, assume that for $\bar{u}$ there exists a path $\bar{u} \to c \to g \to t$, then by construction $\bar{u} \nvDash c$ and $\phi \nvDash c$, since $c \in g$. Hence, F is not satisfied by $\bar{u} \cdot \phi$. Thus, if all nodes from S can reach t, then F is not satisfiable.

The sets d_i can be computed as in the first proof of the section. ◀

A.9 Conditional Lower Bounds For Variable Sensitivity

In this section we prove conditional lower bounds for algorithms where the sensitivity is not fixed, but given a parameter d. Before we give our results, we shortly argue why this setting is relevant.

First, several results were obtained in the setting with sensitivity d. Some of these results are by Patrascu and Thorup [37] for decremental reachability, by Duan and Pettie [23, 24] and by Henzinger and Neumann [28] for subgraph connectivity and by Chechik et al. [19, 18] for decremental all pairs shortest paths. Our lower bounds show that the results in [23, 24] and the incremental algorithm in [28] are tight.

Second, when d is not fixed and we can prove a meaningful lower bound, this will help us understand whether updates or queries are more sensitive to changes of the problem instance.

Third, when d is fixed to a constant, the problems might become easier in the sense that constant or polylogarithmic update and query times can be achieved. For example, for APSP with single edge failures one can achieve query and update times $O(1)$ (see [10]); for APSP with two edge failures one can achieve query and update times $O(\log n)$ (see [22]). In these cases we cannot prove any non-trivial conditional lower bounds for them. However, with an additional parameter d we can derive conditional lower bounds which are polynomial in the parameter d.

Our results are summarized in the following theorem.

▶ **Theorem 21.** *Under the OMv conjecture for any $\varepsilon > 0$, there exists no algorithm with preprocessing time* $\text{poly}(n)$*, update time* $\text{poly}(d)$ *and query time $\Omega(d^{1-\varepsilon})$ for the following problems:*

1. *Decremental/incremental st-SubConn in undirected graphs with sensitivity d*
2. *decremental/incremental st-reach in directed graphs with sensitivity d,*
3. *decremental/incremental BP-Match in undirected bipartite graphs with sensitivity d, and*
4. *decremental/incremental SC in directed graphs with sensitivity d.*
5. *$(2-\varepsilon)$-approximate ss-shortest paths with sensitivity d in undirected unweighted graphs,*
6. *$(5/3-\varepsilon)$-approximate st-shortest paths with sensitivity d in undirected unweighted graphs,*
7. *BW-Matching with sensitivity d.*

Conditional Lower Bounds for Directed Graphs

We observe that some existing reductions can be used to obtain conditional lower bounds for sensitivity problems. In this section, we summarize the results that can be obtained this way.

We call a reduction from a dynamic problem A to another dynamic problem B *sensitivity-preserving* if in the reduction a single update in problem A propagates to problem B as at most one update. We observe that the reductions in [3] from *st*-subgraph-connectivity to *st*-reachability (Lemma 6.1) and from *st*-reachability to BP-Match (Lemma 6.2) and SC (Lemma 6.4) are sensitivity-preserving. Henzinger et al. [27] give conditional lower bounds for the *st*-subgraph-connectivity problem with sensitivity d. The previous observations about sensitivity preserving reductions imply that we get the same lower bounds for st-reach, BP-Match and SC with sensitivity d. This implies the first four points of Theorem 21.

The construction of the lower bound in [27] required $d = m^\delta$ for some $\delta \in (0, 1/2]$. This appears somewhat artifical, since in practice one would rather expect situations with much smaller values for d, e.g., $d = O(1)$ or $d = \text{poly}\log(n)$. However, the lower bound is still interesting because it applies to all algorithms that allow setting $d = m^\delta$. For example, the sensitive subgraph connectivty algorithms of Duan and Pettie [23, 24] is tight w.r.t. to the above lower bound.

Conditional Lower Bounds for Undirected Graphs

In this subsection, we prove the last three points of Theorem 21. We give a reduction from the γ-uMv-problem, which was introduced by Henzinger et al. [27]. The γ-uMv-problem is as follows: Let $\gamma > 0$. An algorithm for the γ-uMv problem is given an $n_1 \times n_2$ binary matrix M with $n_1 = \lfloor n_2^\gamma \rfloor$, that can be preprocessed. Then two vectors u and v appear and the algorithm must output the result of the Boolean vector-matrix-vector-product $u^t M v$.

Henzinger et al. [27, Corollary 2.8] show that under the OMv conjecture for all $\varepsilon > 0$, no algorithm exists for the γ-uMv-problem that has preprocessing time $\text{poly}(n_1, n_2)$, computation time $O(n_1^{1-\varepsilon} n_2 + n_1 n_2^{1-\varepsilon})$, and error probability at most $1/3$. We give a reduction from the γ-uMv-problem to $(2-\varepsilon)$-approximate ss-shortest-paths with sensitivity d in the following lemma. This lemma and the proof of Corollary 3.12 in [27] imply the result in Theorem 21 for ss-shortest-paths.

▶ **Lemma 22.** *Let $\delta \in (0, 1/2]$. Given an algorithm $\mathcal{A}$ for incremental/decremental $(2-\varepsilon)$-ss-shortest paths with sensitivity d, one can solve $(\frac{\delta}{1-\delta})$-uMv with parameters n_1 and n_2 by running the preprocessing step of $\mathcal{A}$ on a graph with $O(m)$ edges and $O(m^{1-\delta})$ vertices, then making a single batch update of size $O(d)$ and $O(m^{1-\delta})$ queries, where m is such that $m^{1-\delta} = n_1$ and $d = m^\delta = n_2$.*

Proof. We prove the lower bound for the incremental problem.

Let M be a $n_1 \times n_2$ binary matrix for $(\frac{\delta}{1-\delta})$-uMv. We construct a bipartite graph G_M from the matrix M: Set $G_M = ((L \cup R), E)$, where $L = \{l_1, \ldots, l_{n_1}\}$, $R = \{r_1, \ldots, r_{n_2}\}$ and the edges are given by $E = \{(l_i, r_j) : M_{ij} = 1\}$. We add an additional vertex s to G_M and attach a path of length 3 to s, the vertex on the path with distance 3 from s has edges to all vertices in L. Observe that G_M has $O(n_1 n_2) = O(m)$ edges and $n_1 + n_2 = \Theta(m^\delta + m^{1-\delta}) = \Theta(m^{1-\delta})$ vertices.

When the vectors u and v arrive, we add an edge (s, r_j) for each $v_j = 1$ in a single batch of $O(d)$ updates. After that for each $u_i = 1$, we query the shortest path from s to l_i. In total, we perform $O(m^{1-\delta})$ queries and only use a single batch update consisting of $O(d)$ insertions. We claim that one of the queries returns less than 4 iff $u^t M v = 1$.

First assume that $u^t M v = 1$. Then there exist indices i, j such that $u_i = M_{ij} = v_j = 1$. Hence, there must be a path $l_i \to r_j \to s$ of length 2 in G_M and since $l_i = 1$ we ask the query for l_i. Hence, any $(2-\varepsilon)$-approximation algorithm must return less than 4 in the query for l_i. Now assume that a query for vertex l_i returns less than 4. Since any path from a vertex in L to s must have length $2k$ for some $k \in \mathbb{N}$, there exists a path $l_i \to r_j \to s$. Since query for l_i, we have $u_i = 1$. Due to the edge $l_i \to r_j$, $M_{ij} = 1$, and due to edge $r_j \to s$, $v_j = 1$. Thus, $u_i M_{ij} v_j = 1$ and $u^t M v = 1$.

The proof for the decremental problem works by initially adding all edges from s to R to the graph and then removing edges corresponding to the 0-entries of v. ◀

To obtain the result of Theorem 21 for st-shortest-paths, observe that the above reduction can be changed to work for this problem: We add additional vertices s, t to the original bipartite graph and connect s and t by a path of length 5 (i.e., introducing 3 additional vertices). Then a similar proof to the above shows that any algorithm for st-reachability that can distinguish between a shortest path of length at most 3 or at least 5 can be used to decide if $u^t M v = 1$. The result for BW-Match follows from the reduction in [3].

B No (globally) fixed constant sensitivity with polynomial preprocessing time

In this section, we will first discuss how SETH reductions depend on their sensitivity. In particular, we will see that even though some of our previous reductions had constant sensitivities, these constants are not bounded globally. Afterwards we will prove that if we allow for polynomial preprocessing time, then there cannot be a globally fixed upper bound on the sensitivities.

B.1 A note on SETH reductions

Let us recall that the sensitivities of the reductions in Section 4 had the form $K = c/\delta$, where cn was the number of clauses in the k-CNFSAT formula and $\delta < 1$ was a parameter indicating the size of our initial graph. We will first discuss the dependency of K on δ and after that on c.

Firstly, let us discuss how the SETH reductions depend on the parameter δ. Let F be a k-CNFSAT formula with n variables and $O(n)$ clauses. Notice that if the input graph that we construct in the reduction had size $N = O(2^{n/2})$, then the preprocessing time of an algorithm refuting SETH would have to be $O(N^{2-\varepsilon})$. In order to allow for arbitrary polynomial preprocessing times of the algorithm, the reductions in [3] (and also the ones from the section before) were parameterized such that the initial graphs have size $O(2^{\delta n})$ (disregarding poly(n) factors). Then for an algorithm with preprocessing time $O(N^t)$ we can pick $\delta < 1/t$ and hence the preprocessing takes time $O(2^{\delta nt}) = O(2^{(1-\gamma)n})$ for some $\gamma > 0$. Thus, despite the "large" preprocessing time we could still refute the SETH. However, the sensitivities $K = c/\delta$ are not bounded if we consider δ as a parameter, i.e. $K \to \infty$ as $\delta \to 0$. If we consider δ as the inverse of the power of the preprocessing time $O(N^t)$, i.e. $\delta = 1/t$, then this can be interpreted as "large preprocessing time" corresponds to "large sensitivity" (since $\delta \to 0$ iff $t \to \infty$).

Now one might want to fix δ (and thus bound the preprocessing time) in order to get (globally) fixed constant sensitivities. Unfortunately, this approach is also not feasible; in the SETH reductions for proving an algorithm that solves k-CNFSAT in time $2^{(1-\varepsilon)n}$, there is another parameter c which denotes the number of clauses of the k-CNFSAT instance after the application of the sparsification lemma by Impagliazzo, Paturi and Zane [30]. Looking at the proof of the lemma one can see that $c = c(k, \varepsilon)$ and that $c \to \infty$ as $k \to \infty$, as well as $c \to \infty$ as $\varepsilon \to 0$. Hence, if we use the sparsification lemma and fix δ, then we still cannot refute the SETH.

B.2 Upper bound in case of polynomial preprocessing

The former reasoning heavily relies on the constructions in our proofs and does not rule the possibility of a reduction with (globally) fixed constant sensitivity from SETH. However, the following theorem and the corollary after it will show that if we allow for polynomial preprocessing time, then there cannot be a polynomial time upper bound on update and query time under any conjecture.

▶ **Theorem 23.** *Let P be a fully dynamic graph problem*[7] *with sensitivity K on a graph*

[7] We only argue about dynamic graph problems to simplify our language. The theorem holds for all dynamic problems with the given properties.

$G = (V, E)$ with n vertices and m edges where edges are added and removed. Assume that for the static version of P there exists an algorithm running in time $\mathrm{poly}(n)$.

Then there exists an algorithm with $p(n) = n^t$ for some $t = t(K)$, $u(n) = O(K)$ and $q(n) = \log(n)$. The algorithm uses space $O(n^t)$.

Proof. The basic idea of the algorithm is to preprocess the results of all possible queries that might be encountered during P. Since we have only sensitivity K, we will only have polynomially many graphs during the running time of the algorithm. We can save all of these results in a balanced tree of height $O(\log n)$ and during a query we just traverse the tree in logarithmic time.

Denote the initial input graph by G_0.

Since we know that after each update has size at most K and after the update we roll back to the initial graph, we can count how many graphs can be created while running P. Particularly, notice that a graph $G_1 = (V, E_1)$ can be created while running P if and only if $S = E_0 \triangle E_1$ has $|S| \leq K$.[8] Then in total the number of graphs which P will have to answer queries on can be bounded by computing how many such sets S exist:

$$\sum_{i=0}^{K} \binom{n^2}{i} \leq K \max_{i=0,\ldots,K} \binom{n^2}{i} = K\,\mathrm{poly}(n) = n^t,$$

for some t (where in the second step we used that K is constant).

Now consider the naïve algorithm which just preprocesses all trees and stores the results: We enumerate all n^t possible trees and run the static algorithm on them. This can be done in time $O(\mathrm{poly}(n))$. We store the results of the static algorithm in a balanced binary tree with $O(n^t)$ leaves which is of height $O(\log n)$. The traversal of the tree can be done, e.g., in the following way: We fix some order $\prec$ on $V \times V$; this implicitly gives an order $\prec$ on the set $\{S \subseteq V \times V\}$. For a graph G_1 we compute $S = E_1 \triangle E_0$ and traverse according to S.

During updates the algorithm maintains an array of size K which contains the edges that are to be removed or added ordered by $\prec$. This can be done in time $O(K) = O(1)$.

During a query the algorithm will traverse the binary tree from the preprocessing according to $\prec$ and the updates that were saved during the updates. This takes time $O(\log n)$.

Hence, we have found algorithm with $p(n) = \mathrm{poly}(n)$ and $u(n) = O(1)$ and $q(n) = O(\log n)$. ◀

▶ **Corollary 24.** *If for a problem with the properties from Theorem 23 there exists a reduction from conjecture $\mathcal{C}$ to P with $p(n) = \mathrm{poly}(n)$ and $\max\{u(n), q(n)\} = \Omega(n^{\gamma-\varepsilon})$ for any $\gamma > 0$ and all $\varepsilon \in (0, \gamma)$. Then $\mathcal{C}$ is false.*

Proof. We use Theorem 23 to obtain an algorithm which is better than the lower bound given in the assumption of the corollary. Hence, we obtain a contradiction to $\mathcal{C}$. ◀

Notice that Corollary 24 implies that in order to obtain meaningful lower bounds for dynamic problems with a certain sensitivity, we must either bound the preprocessing time of the algorithm or bound the space usage of the algorithm or allow the sensitivity to become arbitrarily large.

[8] $A \triangle B$ denotes the symmetric difference of A and B, i.e. $A \triangle B = A \setminus B \cup B \setminus A$.

An Improved Homomorphism Preservation Theorem From Lower Bounds in Circuit Complexity*

Benjamin Rossman

University of Toronto, Canada
ben.rossman@utoronto.ca

Abstract

Previous work of the author [39] showed that the Homomorphism Preservation Theorem of classical model theory remains valid when its statement is restricted to finite structures. In this paper, we give a new proof of this result via a reduction to lower bounds in circuit complexity, specifically on the AC^0 formula size of the colored subgraph isomorphism problem. Formally, we show the following: if a first-order sentence Φ of quantifier-rank k is preserved under homomorphisms on finite structures, then it is equivalent on finite structures to an existential-positive sentence Ψ of quantifier-rank $k^{O(1)}$. Quantitatively, this improves the result of [39], where the upper bound on the quantifier-rank of Ψ is a non-elementary function of k.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes

Keywords and phrases circuit complexity, finite model theory

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.27

1 Introduction

Preservation theorems are a family of results in classical model theory that equate semantic and syntactic properties of first-order formulas. A prominent example — and the subject of this paper — is the Homomorphism Preservation Theorem, which states that a first-order sentence is preserved under homomorphisms if, and only if, it is equivalent to an existential-positive sentence. (Definitions for the various terms in this theorem are given in Section 3.) Two related classical preservation theorems are the Łoś-Tarski Theorem (preserved under *embedding* homomorphisms $\Leftrightarrow$ equivalent to an *existential* sentence) and Lyndon's Theorem (preserved under *surjective* homomorphism $\Leftrightarrow$ equivalent to a *positive* sentence).

In all classical preservation theorems, the "syntactic property $\Rightarrow$ semantic property" direction is straightforward, while the "semantic property $\Rightarrow$ syntactic property" direction is typically proved by an application of the Compactness Theorem.[1] In order to use compactness, it is essential that the semantic property (i.e. preservation under a certain relationship between structures) holds with respect to *all* structures, that is, both finite and infinite. One may also ask about the status of classical preservation theorems relative to a class of structures $\mathscr{C}$. So long as compactness holds in $\mathscr{C}$ (for example, whenever $\mathscr{C}$ is first-order axiomatizable),

* Supported by NSERC and the JST ERATO Kawarabayashi Large Graph Project. This paper was partially written at the National Institute of Informatics in Tokyo and during a visit to IMPA, the National Institute for Pure and Applied Mathematics in Rio de Janeiro.

[1] The Compactness Theorem states that a first-order theory T (i.e. set of first-order sentences) is consistent (i.e. there exists a structure $\mathcal{A}$ which satisfies every sentence in T) if every finite sub-theory of T is consistent. (See [24] for background and proofs of various preservation/amalgamation/interpolation theorems in classical model theory.)

© Benjamin Rossman;
licensed under Creative Commons License CC-BY

so too will all of the classical preservation theorems. The situation is less clear when $\mathscr{C}$ is the class of finite structures (or a subclass thereof), as the Compactness Theorem is easily seen to be false when restricted to finite structures.[2]

The program of classifying theorems in classical model theory according to their validity over finite structures was a major line of research, initiated by Gurevich [20], in the area known as *finite model theory* (see [15, 17, 28]). The status of preservation theorems in particular was systematically investigated in [3, 38]. Given the failure of the Compactness Theorem on finite structures, it is not surprising that nearly all of the classical preservation theorems become false when their statements are restricted to finite structures. A counterexample of Tait [44] from 1959 showed that the Łoś-Tarski Theorem is false over finite structures, while Ajtai and Gurevich [1] in 1987 gave the demise of Lyndon's Theorem via a stronger result in circuit complexity. Namely, they showed that Monotone $\cap$ $\mathrm{AC}^0 \neq$ Monotone-AC^0, that is, there is a (semantically) monotone Boolean function that is computable by AC^0 circuits, but not by (syntactically) monotone AC^0 circuits. The failure of Lyndon's theorem on finite structures follows via the *descriptive complexity* correspondence between AC^0 and first-order logic. (See [26] about the nexus between logics and complexity classes.)

Given the failure of both the Łoś-Tarski and Lyndon Theorems, it might be expected that the Homomorphism Preservation Theorem also fails over finite structures (as it seems to live at the intersection of Łoś-Tarski and Lyndon). On the contrary, however, previous work of the author [39] showed that the Homomorphism Preservation Theorem remains valid over finite structures. The technique of [39] is model-theoretic: its starting point is a new compactness-free proof of the classical theorem, which is then adapted to finite structures. (A summary of the argument is included in Section 8.) In the present paper, we give a completely different proof of this result — and moreover obtain a quantitative improvement — via a reduction to lower bounds in circuit complexity. In particular, we rely on a recent result (of independent interest) that the AC^0 formula size of the colored G-subgraph isomorphism problem is $n^{\Omega(\mathbf{tree\text{-}depth}(G)^{\varepsilon})}$ for an absolute constant $\varepsilon > 0$.

Related Work

Prior to [39], the status of the Homomorphism Preservation Theorem on finite structures was investigated by Feder and Vardi [16], Gräedel and Rosen [18], and Rosen [37], who resolved special cases of the question for restricted classes of first-order sentences. Another special case is due to Atserias [7] in the context of CSP dualities. (See [39] for a discussion of these results.) A different — and incomparable — line of results [6, 14, 31] proves versions of the Homomorphism Preservation Theorems restricted to various *sparse* classes of finite structures (see Ch. 10 of [33], as well as [8] related to the Łoś-Tarski Theorem). See Stolboushkin [43] for an alternative counterexample showing that Lyndon's Theorem fails on finite structures, which is simpler than Ajtai and Gurevich [1] (but doesn't extend to show Monotone-$\mathrm{AC}^0 \neq$ Monotone $\cap$ AC^0).

Outline

The rest of the paper is organized as follows. Because our narrative jumps between logic, graph theory and circuit complexity, for readability sake the various preliminaries — which may be familiar (at least in part) to many readers — are presented in separate sections as

[2] Consider the theory $T = \{\Phi_n : n \in \mathbb{N}\}$ where Φ_n expresses "there exist $\geq n$ distinct elements". Every finite sub-theory of T has a finite model, but T itself does not.

needed throughout the paper. In Section 2, we review basic definitions related to structures, homomorphisms, and first-order logic. In Section 3, we formally state the various preservation theorems discussed in the introduction, including our main result (Theorem 6). Section 4 includes the necessary background on circuit complexity (AC^0 and monotone projections) and graph theory (tree-width, tree-depth, and minor-monotonicity). In Section 5, we introduce the colored G-subgraph isomorphism problem and state the known bounds on its complexity for AC^0 circuits and AC^0 formulas. Section 6 states a needed lemma from descriptive complexity ($FO = AC^0$) and a result connecting quantifier-rank to tree-depth. In Section 7, we prove our main result (Theorem 6) via a reduction to lower bounds for colored G-subgraph isomorphism. (After all the preliminaries, the reduction itself is relatively simple.) For comparison sake, the previous model-theoretic proof technique of [39] is summarized in Section 8. We conclude in Section 9 with a brief discussion of syntax vs. semantics in circuit complexity.

2 Preliminaries, I

2.1 Structures and Homomorphisms

Throughout this paper, let σ be a fixed finite relational signature, that is, a list of relation symbols $R^{(r)}$ (where $r \in \mathbb{N}$ denotes the arity of R). A *structure* $\mathcal{A}$ consists of a set A (called the unverse of $\mathcal{A}$) together with interpretations $R^{\mathcal{A}} \subseteq A^r$ for each relation symbol $R^{(r)}$ in σ. A priori, structures may be finite or infinite.

A *homomorphism* from a structure $\mathcal{A}$ to a structure $\mathcal{B}$ is a map $f : A \to B$ such that $(a_1, \ldots, a_r) \in R^{\mathcal{A}} \implies (f(a_1), \ldots, f(a_r)) \in R^{\mathcal{B}}$ for every $R^{(r)} \in \sigma$ and $(a_1, \ldots, a_r) \in A^r$. Notation $\mathcal{A} \to \mathcal{B}$ asserts the existence of a homomorphism from $\mathcal{A}$ to $\mathcal{B}$.

A homomorphism $f : \mathcal{A} \to \mathcal{B}$ is an *embedding* if f is one-to-one and satisfies $(a_1, \ldots, a_r) \in R^{\mathcal{A}} \iff (f(a_1), \ldots, f(a_r)) \in R^{\mathcal{B}}$ for every $R^{(r)} \in \sigma$ and $(a_1, \ldots, a_r) \in A^r$.

2.2 First-Order Logic

First-order formulas (in the relational signature σ) are constructed out of atomic formulas (of the form $x_1 = x_2$ or $R(x_1, \ldots, x_r)$ where $R^{(r)} \in \sigma$ and x_i's are variables) via boolean connectives ($\varphi \wedge \psi$, $\varphi \vee \psi$, and $\neg\varphi$) and universal and existential quantification ($\forall x\ \varphi(x)$ and $\exists x\ \varphi(x)$). For a structure $\mathcal{A}$ and a first-order formula $\varphi(x_1, \ldots, x_k)$ and a tuple of elements $\vec{a} \in A^k$, notation $\mathcal{A} \models \varphi(\vec{a})$ is the statement that $\mathcal{A}$ satisfies φ with $\vec{a}$ instantiating the free variables $\vec{x}$. First-order formulas with no free variables are called *sentences* and represented by capital Greek letters Φ and Ψ.

A first-order sentence (or formula) is said to be:

- *positive* if it does not contain any negations (that is, it has no sub-formula of the form $\neg\varphi$),
- *existential* if it contains only existential quantifiers (that is, it has no universal quantifiers) and has no negations outside the scope of any quantifier, and
- *existential-positive* if it is both existential and positive.

Two important parameters first-order sentences are quantifier-rank and variable-width. *Quantifier-rank* is the maximum nesting depth of quantifiers. *Variable-width* is the maximum number of free variables in a sub-formula. As we will see in Section 6, under the descriptive complexity characterization of first-order logic in terms of AC^0 circuits, variable-width corresponds to AC^0 circuit size and quantifier-rank corresponds to AC^0 formula size (or, more accurately, AC^0 formula depth when fan-in is restricted to $O(n)$).

Note that first-order sentences are not assumed to be in prenex form. For example, the formula $(\exists x\ P(x)) \vee (\exists y\ \neg Q(y))$ is existential (but not positive) and has quantifier-rank 1 and variable-width 1.

3 The Homomorphism Preservation Theorem

▶ **Definition 1.** A first-order sentence Φ is *preserved under homomorphisms* [*on finite structures*] if $(\mathcal{A} \models \Phi$ and $\mathcal{A} \to \mathcal{B}) \Longrightarrow \mathcal{B} \models \Phi$ for all [finite] structures $\mathcal{A}$ and $\mathcal{B}$. The notions of *preserved under embeddings* and *preserved under surjective homomorphisms* are defined similarly.

We now formally state the three classical preservations mentioned in the introduction.

▶ **Theorem 2** (Łoś-Tarski / Lyndon / Homomorphism Preservation Theorems [29])**.** *A first-order sentence is preserved under* [*embedding / surjective / all*] *homomorphisms if, and only if, it is equivalent to an* [*existential / positive / existential-positive*] *sentence.*

As discussed in the introduction, Łoś-Tarski and Lyndon's Theorems become false when restricted to finite structures.

▶ **Theorem 3** (Failure of Łoś-Tarski and Lyndon Theorems on Finite Structures [1, 44])**.** *There exists a first-order sentence that is preserved under* [*embedding / surjective*] *homomorphisms on finite structures, but is not equivalent on finite structures to any* [*existential / positive*] *sentence.*

In contrast, the Homomorphism Preservation Theorem remains valid over finite structures.

▶ **Theorem 4** (Homomorphism Preservation Theorem on Finite Structures [39])**.** *If a first-order sentence of quantifier-rank k is preserved under homomorphisms on finite structures, then it is equivalent on finite structures to an existential-positive sentence of quantifier-rank $\beta(k)$, for some computable function $\beta : \mathbb{N} \to \mathbb{N}$.*

We will refer to $\beta : \mathbb{N} \to \mathbb{N}$ in Theorem 4 as the "quantifier-rank blow-up". (Formally, there is one computable function $\beta_\sigma : \mathbb{N} \to \mathbb{N}$ for each finite relational signature σ.) We remark that the upper bound on $\beta(k)$ given by the proof of Theorem 4 is a non-elementary function of k (i.e. it is grows faster than any bounded-height tower of exponentials). In contrast, a second result in [39] shows that the optimal bound $\beta(k) = k$ holds in the classical Homomorphism Preservation Theorem.

▶ **Theorem 5** ("Equi-rank" Homomorphism Preservation Theorem [39])**.** *If a first-order sentence of quantifier-rank k is preserved under homomorphism, then it is equivalent to an existential-positive sentence of quantifier-rank k.*

Due to reliance on the Compactness Theorem, the original proof of the classical Homomorphism Preservation Theorem gives no computable upper bound whatsoever on the quantifier-rank blow-up. Theorem 5 is proved by a constructive, compactness-free argument (see Section 8). In [39] I conjectured that this stronger "equi-rank" theorem is valid over finite structures. However, new techniques were clearly needed to improve the non-elementary upper bound on $\beta(k)$.

The main result of the present paper is a completely new proof of Theorem 4, which moreover gives a polynomial upper bound on $\beta(k)$.

▶ **Theorem 6** ("Poly-rank" Homomorphism Preservation Theorem on Finite Structures)**.** *If a first-order sentence of quantifier-rank k is preserved under homomorphisms on finite structures, then it is equivalent on finite structures to an existential-positive sentence of quantifier-rank $k^{O(1)}$.*

The proof of Theorem 6 involves a reduction to the AC^0 formula size of SUB_G, the colored G-subgraph isomorphism problem. This reduction transforms lower bounds on the AC^0 formula size of SUB_G into upper bounds on the quantifier-rank blow-up $\beta(k)$ in Theorem 4. In Section 7.1, we derive an exponential upper bound $\beta(k) \leq 2^{O(k)}$ from an existing lower bound of [41] on the AC^0 formula size of SUB_{P_k} (also known as the distance-k connectivity problem). Two further steps, described in Section 7.2, are required for the polynomial upper bound $\beta(k) \leq k^{O(1)}$ of Theorem 6. The first is a new result in graph minor theory from [25] (joint work with Ken-ichi Kawarabayashi), which gives a "polynomial excluded-minor approximation" of tree-depth, analogous to the Polynomial Grid-Minor Theorem of Chekuri and Chuzhoy [12]. The second ingredient, in a forthcoming paper of the author [42], is a lower bound on AC^0 formula size of SUB_G in the special case where G is complete binary tree.

4 Preliminaries, II

4.1 Circuit Complexity

We consider *Boolean circuits* with unbounded fan-in AND and OR gates and negations on inputs. That is, inputs are labelled by variables x_i or negated variables $\overline{x}_i$ (where i comes from some finite index set, typically $\{1, \ldots, n\}$). We measure *size* by the number of gates and *depth* by the maximum number of gates on an input-to-output path. Boolean circuits with fan-out 1 (i.e. tree-like Boolean circuits) are called *Boolean formulas*. (Boolean formulas are precisely the same as quantifier-free first-order formulas.)

The *depth-d circuit/formula size* of a Boolean function f is the minimum size of a depth-d circuit/formula that computes f. AC^0 refers to constant-depth, $\mathrm{poly}(n)$-size sequences of Boolean circuits/formula on $\mathrm{poly}(n)$ variables. For a sequence (f_n) of Boolean functions on $\mathrm{poly}(n)$ variables and a constant $c > 0$, we say that "(f_n) has AC^0 circuit/formula size $O(n^c)$ (resp. $\Omega(n^c)$)" if for some d (resp. for all d), the depth-d circuit/formula size of f_n is $O_d(n^c)$ (resp. $\Omega_d(n^c)$) for all n.

One slightly unusual complexity measure (which arises in the descriptive complexity correspondence between AC^0 and first-order logic in Section 6) is *fan-in n depth*, that is, the minimum depth required to compute a Boolean function by AC^0 circuits with fan-in restricted to n. Note that AC^0 formula size lower bounds imply fan-in n depth lower bounds: if f has AC^0 formula size $\omega(n^c)$, then its fan-in n formula depth is at least c (for sufficiently large n). (This follows from the observation that every depth-d formula with fan-in n is equivalent to a depth-d formula of size at most n^d.)

4.2 Monotone Projections

▶ **Definition 7** (Monotone-Projection Reductions)**.** For Boolean functions $f : \{0,1\}^I \to \{0,1\}$ and $g : \{0,1\}^J \to \{0,1\}$, a *monotone-projection reduction* from f to g is a map $\rho : J \to$

$I \cup \{0,1\}$ such that $f(x) = g(\rho^*(x))$ for all $x \in \{0,1\}^I$ where $\rho^*(x) \in \{0,1\}^J$ is defined by

$$(\rho^*(x))_j = \begin{cases} x_i & \text{if } \rho(j) = i \in I, \\ 0 & \text{if } \rho(j) = 0, \\ 1 & \text{if } \rho(j) = 1. \end{cases}$$

(Properly speaking, the "reduction" from f to g is the map $\rho^* : \{0,1\}^I \to \{0,1\}^J$ induced by ρ.) Notation $f \leq_{\text{mp}} g$ denotes the existence of a monotone-projection reduction from f to g.

When describing monotone-projection reductions later in this paper, it will be natural to speak in terms of indexed sets of Boolean variables $\{X_i\}_{i \in I}$ and $\{Y_j\}_{j \in J}$, rather than sets I and J themselves. Thus, a monotone-projection reduction $\rho : J \to I \cup \{0,1\}$ associates each variable Y_j with either a constant (0 or 1) or some variable X_i.

Note that $\leq_{\text{mp}}$ is a partial order on Boolean functions. This is the simplest kind of reduction in complexity theory. It has the nice property that every standard complexity measure on Boolean functions is monotone under $\leq_{\text{mp}}$. For instance, letting $L_d(f)$ denote the depth-d formula size of f, we have $f \leq_{\text{mp}} g \implies L_d(f) \leq L_d(g)$.

4.3 Tree-Width and Tree-Depth

Graphs in this paper are finite simple graphs. (In contrast to the previous discussion of infinite structures, we assume finiteness whenever we speak of graphs.) Formally, a graph G is a pair $(V(G), E(G))$ where $V(G)$ is a finite set and $E(G) \subseteq \binom{V(G)}{2}$ is a set of unordered pairs of vertices.

Four specific graphs that arise in this paper: for $k \geq 1$, let K_k denote the complete graph of order k, let P_k denote the path of order k, let B_k denote the complete binary tree of height k (where every leaf-to-root path has order k), and let $Grid_{k \times k}$ denote the $k \times k$ grid graph. (In the case $k = 1$, all four of these graphs are a single vertex.)

We recall the definitions of two structural parameters, tree-width and tree-depth, which play an important role in this paper. A *tree decomposition* of a graph G consists of a tree T and a family $\mathcal{W} = \{W_t\}_{t \in V(T)}$ of sets $W_t \subseteq V(G)$ satisfying

- $\bigcup_{t \in V(T)} W_t = V(G)$ and every edge of G has both ends in some W_t, and
- if $t, t', t'' \in V(T)$ and t' lies on the path in T between t and t'', then $W_t \cap W_{t''} \subseteq W_{t'}$.

The *tree-width* of G, denoted $\mathbf{tw}(G)$, is the minimum of $\max_{t \in V(T)} |W_t| - 1$ over all tree decompositions $(T, \mathcal{W})$ of G.

The *tree-depth* of G, denoted $\mathbf{td}(G)$, is the minimum height of a rooted forest F such that $V(F) = V(G)$ and every edge of G has both ends in some branch in F (i.e. for every $\{v, w\} \in E(G)$, vertices v and w have an ancestor-descendant relationship in F). There is also an inductive characterization of tree-depth: if G has connected components $G_1, \ldots, G_t$, then

$$\mathbf{td}(G) = \begin{cases} 1 & \text{if } |V(G)| = 1, \\ 1 + \min\limits_{v \in V(G)} \mathbf{td}(G - v) & \text{if } t = 1 \text{ and } |V(G)| > 1, \\ \max\limits_{i \in \{1,\ldots,t\}} \mathbf{td}(G_i) & \text{if } t > 1. \end{cases}$$

These two structural parameters, tree-width and tree-depth, are related by inequalities:

$$\mathbf{tw}(G) \leq \mathbf{td}(G) - 1 \leq \mathbf{tw}(G) \cdot \log |V(G)|. \tag{1}$$

Tree-depth is also related the length of the longest path in G, denoted $\mathbf{lp}(G)$:

$$\log(\mathbf{lp}(G)+1) \leq \mathbf{td}(G) \leq \mathbf{lp}(G). \tag{2}$$

(See Ch. 6 of [33] for background on tree-depth and proofs of these inequalities.)

Graph parameters $\mathbf{tw}(\cdot)$ and $\mathbf{td}(\cdot)$, as well as $\mathbf{lp}(\cdot)$, are easily seen to be monotone under the graph-minor relation. A reminder what this means: recall that a graph H is a *minor* of a graph G, denoted $H \preceq G$, if H can be obtained from G by a sequence of edge contractions and vertex/edge deletions. A graph parameter $f : \{\text{graphs}\} \to \mathbb{N}$ is said to be *minor-monotone* if $H \preceq G \implies f(H) \leq f(G)$ for all graphs H and G.

5 The Colored G-Subgraph Isomorphism Problem

In this section, we introduce the colored G-subgraph isomorphism problem and state the known upper and lower bounds on its complexity with respect to AC^0 circuits and formulas.

▶ **Definition 8.** For a graph G and $n \in \mathbb{N}$, the *blow-up* $G^{\uparrow n}$ is the graph defined by

$$V(G^{\uparrow n}) = V(G) \times [n],$$
$$E(G^{\uparrow n}) = \big\{\{(v,a),(w,b)\} : \{v,w\} \in E(G),\ a,b \in [n]\big\}.$$

For $\alpha \in [n]^{V(G)}$, let $G^{(\alpha)}$ denote the subgraph of $G^{\uparrow n}$ defined by

$$V(G^{(\alpha)}) = \big\{(v,\alpha_v) : v \in V(G)\big\},$$
$$E(G^{(\alpha)}) = \big\{\{(v,\alpha_v),(w,\alpha_w)\} : \{v,w\} \in E(G)\big\}.$$

(Note that each $G^{(\alpha)}$ is an isomorphic copy of G.)

▶ **Definition 9.** For any fixed graph G, the *colored G-subgraph isomorphism problem* asks, given a subgraph $X \subseteq G^{\uparrow n}$, to determine whether or not there exists $\alpha \in [n]^{V(G)}$ such that $G^{(\alpha)} \subseteq X$. For complexity purposes, we view this problem as a Boolean function $\mathrm{SUB}_{G,n} : \{0,1\}^{|E(G)|\cdot n^2} \to \{0,1\}$ with variables $\{X_e\}_{e \in E(G^{\uparrow n})}$. We write SUB_G for the sequence of Boolean functions $\{\mathrm{SUB}_{G,n}\}_{n\in\mathbb{N}}$.

5.1 Minor-Monotonicity

The following observation appears in [27].

▶ **Proposition 10.** *If H is a minor of G, then $\mathrm{SUB}_H \leq_{\mathrm{mp}} \mathrm{SUB}_G$ (i.e. $\mathrm{SUB}_{H,n} \leq_{\mathrm{mp}} \mathrm{SUB}_{G,n}$ for all $n \in \mathbb{N}$).*

Proof. By transitivity of $\leq_{\mathrm{mp}}$, it suffices to consider the two cases where H is obtained from G via deleting or contracting a single edge $\{v,w\} \in E(G)$. In both cases, the monotone projection maps each variable $X_{\{(v',a),(w',b)\}}$ of SUB_G with $\{v',w'\} \neq \{v,w\}$ to the correspond variable $Y_{\{(v',a),(w',b)\}}$ of SUB_H. In the deletion case, we set the variable $X_{\{(v,a),(w,b)\}}$ to the constant 1 for all $a,b \in [n]$. In the contraction case, we set $X_{\{(v,a),(w,b)\}}$ to 1 if $a = b$ and to 0 if $a \neq b$. (This "planted perfect matching" has the effect of gluing the v-fibre and the w-fibre for instances of SUB_H.) ◀

Proposition 10 implies that the graph parameter $G \mapsto \mu(\mathrm{SUB}_G)$ is minor-monotone for any standard complexity measure $\mu : \{\text{Boolean functions}\} \to \mathbb{N}$ (e.g. depth-d AC^0 formula size). It also implies:

▶ **Corollary 11.** *For all graphs G,* $\mathrm{SUB}_{P_{\mathbf{td}(G)}} \leq_{\mathrm{mp}} \mathrm{SUB}_G$.

Proof. Recall that $\mathbf{td}(G) \leq \mathbf{lp}(G)$ by inequality (2). That is, every graph G contains a path of length $\mathbf{td}(G)$.[3] Since subgraphs are minors, we have $P_{\mathbf{td}(G)} \preceq G$ and therefore $\mathrm{SUB}_{P_{\mathbf{td}(G)}} \leq_{\mathrm{mp}} \mathrm{SUB}_G$ by Proposition 10. ◀

5.2 Upper Bounds

The obvious "brute-force" way of solving SUB_G has running time $O(n^{|V(G)|})$: given an input $X \subseteq G^{\uparrow n}$, check if $G^{(\alpha)} \subseteq X$ for each $\alpha \in [n]^{V(G)}$. A better upper bound comes from tree-width: based on an optimal tree-decomposition $(T, \mathcal{W})$, there is a dynamic-programming algorithm with running time $n^{\mathbf{tw}(G)+O(1)}$ [34]. This algorithm can be implemented by AC^0 circuits of size $n^{\mathbf{tw}(G)+O(1)}$ and depth $O(|V(G)|)$.[4]

Unlike circuits, formulas cannot faithfully implement dynamic-programming algorithms. The fastest known formulas for SUB_G are tied to tree-depth: based on a minimum-height rooted forest F witnessing $\mathbf{td}(G)$, there are AC^0 formulas of size $n^{\mathbf{td}(G)+O(1)}$ solving SUB_G (which come from AC^0 circuits of depth $\mathbf{td}(G)+O(1)$ and fan-in $O(n)$). For future reference, these upper bounds are stated in the following proposition.[5]

▶ **Proposition 12.** *For all graphs G,* SUB_G *is solvable by* AC^0 *circuits of size* $n^{\mathbf{tw}(G)+O(1)}$, *as well as by* AC^0 *formulas of size* $n^{\mathbf{td}(G)+O(1)}$.

5.3 Lower Bounds: AC^0 Circuit Size

Previous work of the author [40] showed that the AC^0 circuit size of SUB_{K_k} (a.k.a. the (colored) k-CLIQUE problem) is $n^{\Omega(k)}$ for every $k \in \mathbb{N}$. Generalizing the technique of [40], Amano [5] gave a lower bound on the AC^0 circuit size of SUB_G for arbitrary graphs G. In particular, he showed that the AC^0 circuit size of $\mathrm{SUB}_{Grid_{k\times k}}$ is $n^{\Omega(k)}$. This result, combined with the recent Polynomial Grid-Minor Theorem[6] of Chekuri and Chuznoy [12], implies that the AC^0 circuit size of SUB_G is $n^{\Omega(\mathbf{tw}(G)^{\varepsilon})}$ for an absolute constant $\varepsilon > 0$. An even stronger lower bound was subsequently proved by Li, Razborov and Rossman [27] (without appealing to the Polynomial Grid-Minor Theorem).

▶ **Theorem 13.** *For all graphs G, the* AC^0 *circuit size of* SUB_G *is* $n^{\Omega(\mathbf{tw}(G)/\log \mathbf{tw}(G))}$.

This result is nearly tight, as it matches the upper bound of Proposition 12 up to the $O(\log \mathbf{tw}(G))$ factor in the exponent.

---

[3] This fact is straightforward to prove. Consider the case that G is connected. Starting at any vertex of G, constructed a rooted tree T by a depth-first search. Observe that for every edge $\{v, w\} \in E(G)$, it must be the case that v and w lie in a common branch of T. Therefore, the height of T is an upper bound on $\mathbf{td}(G)$. On the other hand, note that each root-to-leaf branch of T is a path in G. Therefore, the height of T is a lower bound on $\mathbf{lp}(G)$.

[4] It may be possible to achieve running times of $n^{\delta \cdot \mathbf{tw}(G)+O(1)}$ for constants $\delta < 1$ using fast matrix multiplication algorithms (cp. [46]). However, these algorithms appear to require logarithmic-depth circuits. For unrestricted Boolean circuits, no upper bound better than $n^{O(\mathbf{tw}(G))}$ is known, and in fact Marx [30] has shown that the Strong Exponential Time Hypothesis rules out circuits smaller than $n^{O(\mathbf{tw}(G)/\log \mathbf{tw}(G))}$.

[5] For the *uncolored* G-subgraph isomorphism graph, one gets essentially the same upper bounds via a reduction to SUB_G using the "color-coding" technique of Alon, Yuster and Zwick [4]. Amano [5] observed that this uncolored-to-colored reduction can be implemented by AC^0 circuits.

[6] This states every graph G of tree-width k contains an $\Omega(k^{\varepsilon}) \times \Omega(k^{\varepsilon})$ grid minor for an absolute constant $\varepsilon > 0$.

5.4 Lower Bounds: AC^0 Formula Size

For the main result of this paper (Theorem 6), we require a lower bound on the AC^0 formula size of SUB_G (or in fact on the fan-in $O(n)$ depth of SUB_G). Since formulas are a subclass of circuits, Theorem 13 implies that the AC^0 formula size of SUB_G is at least $n^{\Omega(\mathbf{tw}(G)/\log \mathbf{tw}(G))}$. However, this does not match the $n^{\mathbf{td}(G)+O(1)}$ lower bound of Proposition 12, since $\mathbf{td}(G)$ may be larger than $\mathbf{tw}(G)$ (by up to a $\log|V(G)|$ factor). In particular, the path P_k has tree-width 1 and tree-depth $\lceil \log(k+1) \rceil$. Although Theorem 13 gives no non-trivial lower bound on the AC^0 formula size of SUB_{P_k}, a nearly optimal lower bound was proved in different work of the author [41]:

▶ **Theorem 14.** *The* AC^0 *formula size of* SUB_{P_k} *is* $n^{\Omega(\log k)}$. *More precisely, the depth-*d *formula size of* $\mathrm{SUB}_{P_k,n}$ *is* $n^{\Omega(\log k)}$ *for all* $k, d, n \in \mathbb{N}$ *with* $k \leq \log\log n$ *and* $d \leq \frac{\log n}{(\log\log n)^3}$.

Via the relationship between AC^0 formula size and fan-in $O(n)$ circuit depth, Theorem 14 implies:

▶ **Corollary 15.** *Circuits with fan-in* $O(n)$ *computing* SUB_{P_k} *have depth* $\Omega(\log k)$.

▶ **Remark 16.** We mention a few other lower bounds related to Corollary 15. A recent paper of Chen, Oliveira, Servedio and Tan [13] gives a nearly optimal size-depth trade-off for AC^0 circuits computing SUB_{P_k}. Namely, they prove that the depth-d circuit size of $\mathrm{SUB}_{P_k,n}$ is $n^{\Omega(d^{-1}k^{1/(d-1)})}$ for all $k \leq n^{1/5}$. (This result is incomparable to Theorem 14.) As a corollary, this shows that circuits with fan-in $O(n)$ computing SUB_{P_k} have depth $\Omega(\log k/\log\log k)$ (a slightly weaker bound than Corollary 15). Previous size-depth trade-offs due to Beame, Impagliazzo and Pitassi [9] and Ajtai [2] imply lower bounds of $\Omega(\log\log k)$ and $\Omega(\log^* k)$ respectively on the fan-in $O(n)$ depth of SUB_{P_k}.

In Section 7.1, we use Corollary 15 (together with Corollary 11) to prove a weak version of our main result, Theorem 6, with an exponential upper bound $\beta(k) \leq 2^{O(k)}$ on the quantifier-rank blow-up. We remark that the lower bound of Chen et al. implies a slightly weaker upper bound of $k^{O(k)}$, while the very first non-trivial lower bound of Ajtai implies a *non-elementary* upper bound on $\beta(k)$ (similar to the original proof of Theorem 4). For the polynomial upper bound $\beta(k) \leq k^{O(1)}$, we require a stronger $n^{\Omega(\mathbf{td}(G)^{\varepsilon})}$ lower bound on the AC^0 formula size of SUB_G for arbitrary graphs G, as we explain in Section 7.2.

6 Preliminaries, III

In this section, we state a few needed lemmas on the relationship between first-order logic and AC^0 formula size. As before, let σ be a fixed finite relational signature. However, we now stipulate that ***all structures in Sections 6 and 7 are finite***. That is, we drop the adjective "finite" everywhere since it is assumed. Asymptotic notation in these sections ($O(\cdot)$, etc.) implicitly depends on σ (although, essentially without loss of generality, it suffices to prove our results in the special case $\sigma = \{R^{(2)}\}$ of a single binary relation).

6.1 Descriptive Complexity: $\mathrm{FO} = \mathrm{AC}^0$

▶ **Definition 17** (Gaifman Graphs, Encodings, MODEL_Φ).

- For a structure $\mathcal{A}$, we denote by $\mathrm{Gaif}(\mathcal{A})$ the *Gaifman graph* of $\mathcal{A}$. This is the graph whose vertex set is the universe of $\mathcal{A}$ and whose edges are pairs $\{v, w\}$ such that $v \neq w$ and v, w appear together in a tuple of any relation of $\mathcal{A}$.

- If $\mathcal{A}$ has universe $[n]$, then we denote by $\mathrm{Enc}(\mathcal{A}) \in \{0,1\}^{\widehat{n}}$ the standard bit-string encoding of $\mathcal{A}$ where $\widehat{n} = \sum_{R^{(t)} \in \sigma} n^t$ $(= n^{O_\sigma(1)})$. That is, each bit of $\mathrm{Enc}(\mathcal{A})$ is the indicator for a tuple of some relation of $\mathcal{A}$. (Note that $\mathrm{Enc}(\cdot)$ is a bijection between structures with universe $[n]$ and strings in $\{0,1\}^{\widehat{n}}$.)
- For a first-order sentence Φ and $n \in \mathbb{N}$, let $\mathrm{MODEL}_{\Phi,n} : \{0,1\}^{\widehat{n}} \to \{0,1\}$ be the Boolean function defined, for structures $\mathcal{A}$ with universe $[n]$, by

$$\mathrm{MODEL}_{\Phi,n}(\mathrm{Enc}(\mathcal{A})) = 1 \overset{\text{def}}{\iff} \mathcal{A} \models \Phi.$$

We write MODEL_Φ for the sequence of Boolean functions $\{\mathrm{MODEL}_{\Phi,n}\}_{n\in\mathbb{N}}$.

The next lemma gives one-half of the descriptive complexity correspondence between first-order logic and AC^0:

▶ **Lemma 18** ("$\mathrm{FO} \subseteq \mathrm{AC}^0$"). *For all $1 \le w \le k$, if Φ is a first-order sentence of quantifier-rank k and variable-width w, then MODEL_Φ is computable by AC^0 circuits of depth k and fan-in $O(n)$ and size $O(n^w)$. These circuits are equivalent with AC^0 formulas of depth k and size $O(n^k)$.*

(To be completely precise, each of these $O(\cdot)$ terms is really $O_{\sigma,k}(\cdot)$, that is, with constants that depend on k as well as the signature σ.) We remark that Lemma 18 has a converse ("$\mathrm{AC}^0 \subseteq \mathrm{FO}$") with respect to both the uniform and non-uniform versions of AC^0. We omit the statement of these results, since the description of AC^0 circuits via first-order sentences is not needed in this paper (see [26] for details).

6.2 Retracts, Cores, Hom-Preserved Classes

The last bit of required background concerns homomorphism-preserved classes of structures. We begin by defining the key notions of homomorphic equivalence and cores.

▶ **Definition 19** (Homomorphic Equivalence, (Co-)Retracts, Cores).

- Recall notation $\mathcal{A} \to \mathcal{B}$ denoting the existence of a homomorphism from $\mathcal{A}$ to $\mathcal{B}$.
- Structures $\mathcal{A}$ and $\mathcal{B}$ are *homomorphically equivalent*, denoted $\mathcal{A} \rightleftarrows \mathcal{B}$, if $\mathcal{A} \to \mathcal{B}$ and $\mathcal{B} \to \mathcal{A}$.
- We write $\mathcal{A} \overset{\supseteq}{\to} \mathcal{B}$ and say that $\mathcal{B}$ is a *retract* of $\mathcal{A}$ and $\mathcal{A}$ is a *co-retract* of $\mathcal{B}$ if: (1) $\mathcal{B}$ is a substructure of $\mathcal{A}$ and (2) there exists a homomorphism $\mathcal{A} \to \mathcal{B}$ that fixes $\mathcal{B}$ pointwise (a.k.a. a retraction). (Note that $\mathcal{A} \overset{\supseteq}{\to} \mathcal{B}$ implies $\mathcal{A} \rightleftarrows \mathcal{B}$.)
- A structure $\mathcal{A}$ is a *core* if it has no proper retract (that is, $\mathcal{A} \overset{\supseteq}{\to} \mathcal{B} \implies \mathcal{A} = \mathcal{B}$).

The next lemma states a few basic properties of cores (see [22, 23]).

▶ **Lemma 20.**

(a) *Every $\rightleftarrows$-equivalence class contains a unique core up to isomorphism. (That is, every structure $\mathcal{A}$ is homomorphically equivalent to a unique core.)*

(b) *For every k, there are only finitely many non-isomorphic cores of tree-depth k. (This number depends on the signature σ.)*

(c) *(As an aside:) If a graph G is a core, then the colored and uncolored G-subgraph isomorphism problems are equivalent under linear-size monotone-projection reductions (see [19, 27]).*

▶ **Definition 21** (Hom-Preserved Classes, Minimal Cores).

- We say that a class of structures $\mathscr{C}$ (i.e. a class of finite structures) is *hom-preserved* if, whenever $\mathcal{A} \in \mathscr{C}$ and $\mathcal{A} \to \mathcal{B}$, we have $\mathcal{B} \in \mathscr{C}$.

- For a hom-preserved class $\mathscr{C}$, let MinCores($\mathscr{C}$) be the set of $\mathcal{M} \in \mathscr{C}$ with the property that for all structures $\mathcal{A}$, if $\mathcal{A} \in \mathscr{C}$ and $\mathcal{A} \to \mathcal{M}$, then $\mathcal{M}$ is isomorphic to a retract of $\mathcal{A}$.

The next lemma states the essential properties of MinCores($\mathscr{C}$) (see [39]).

▶ **Lemma 22.** *The following hold for any hom-preserved class $\mathscr{C}$:*

(a) *$\mathcal{A} \in \mathscr{C}$ if, and only if, there exists $\mathcal{M} \in$ MinCores($\mathscr{C}$) such that $\mathcal{M} \to \mathcal{A}$.*

(b) *Every structure in* MinCores($\mathscr{C}$) *is, indeed, a core.*

(c) *Every homomorphism between structures in* MinCores($\mathscr{C}$) *is an isomorphism.*

(d) *$\mathscr{C}$ is definable (i.e. within the class of all finite structures) by an existential-positive sentence of quantifier-rank k if, and only if, $\mathbf{td}(\text{Gaif}(\mathcal{M})) \leq k$ for all $\mathcal{M} \in$ MinCores($\mathscr{C}$).*

(e) *$\mathscr{C}$ is definable by an existential-positive sentence of variable-width w if, and only if,* MinCores($\mathscr{C}$) *contains finitely many non-isomorphic structures and $\mathbf{tw}(\text{Gaif}(\mathcal{M})) \leq w$ for every $\mathcal{M} \in$ MinCores($\mathscr{C}$).*

Since Lemma 22(d) in particular plays a key role in the next section, we briefly sketch the proof. In one direction: Suppose $\mathscr{C}$ is defined by an existential-positive sentence Φ of quantifier-rank k. It is easy to show (by a syntactic argument) that Φ is equivalent to a disjunction $\Psi_1 \vee \cdots \vee \Psi_t$ of *primitive-positive sentences* Ψ_i (i.e. existential-positive sentences that involve conjunctions $\wedge$ but no disjunctions $\vee$), each with quantifier-rank at most k. For each Ψ_i, there is a corresponding structure $\mathcal{A}_i$ with the property that $\mathcal{B} \models \Psi_i \Leftrightarrow \mathcal{A}_i \to \mathcal{B}$ and moreover the tree-depth of $\mathcal{A}_i$ is at most the quantifier-rank of Ψ_i (and hence at most k). Thus, $\mathscr{C}$ is generated by $\mathcal{A}_1, \ldots, \mathcal{A}_t$ and hence MinCores($\mathscr{C}$) consists of finitely many cores, each of tree-depth at most k (coming from the minimal elements among $\mathcal{A}_1, \ldots, \mathcal{A}_t$ in the homomorphism order).

For the reverse direction: Start with the assumption that all structures in MinCores($\mathscr{C}$) have tree-depth at most k. By Lemma 20(b), MinCores($\mathscr{C}$) contains finitely many non-isomorphic structures $\mathcal{M}_1, \ldots, \mathcal{M}_t$. For each $\mathcal{M}_i$, let Ψ_i be the corresponding primitive-positive sentence of quantifier-rank at most k. Then $\mathscr{C}$ is defined by the existential-positive sentence $\Psi_1 \vee \cdots \vee \Psi_t$.

7 Proof of Theorem 6

In this section, we finally prove our main result, the "Poly-rank" Homomorphism Preservation Theorem on Finite Structures (Theorem 6, stated in Section 3). We begin in Section 7.1 by proving a weaker version of the result with an exponential upper bound $\beta(k) \leq 2^{O(k)}$. In Section 7.2, we describe the improvement to $\beta(k) \leq k^{O(1)}$, which involves new results from circuit complexity and graph minor theory.

7.1 Preliminary Bound: $\beta(k) \leq 2^{O(k)}$

For simplicity sake, we will assume that σ consists of binary relations only. At the end of this subsection, we explain how to extend the argument to arbitrary σ.

Let Φ be a first-order sentence of quantifier-rank k, let $\mathscr{C}$ be the set of finite models of Φ, and assume that $\mathscr{C}$ is hom-preserved (that is, Φ is preserved under homomorphisms on finite structures). Our goal is to show that Φ is equivalent to an existential-positive sentence of quantifier-rank $2^{O(k)}$. By Lemma 22(d), it suffices to show that $\mathbf{td}(\text{Gaif}(\mathcal{M})) \leq 2^{O(k)}$ for all $\mathcal{M} \in$ MinCores($\mathscr{C}$).

Consider any $\mathcal{M} \in$ MinCores($\mathscr{C}$). Let G be the Gaifman graph of $\mathcal{M}$, and let m be the size of the universe of $\mathcal{M}$. (Note that $m = |V(G)|$.) The following claim is key to showing $\mathbf{td}(G) \leq 2^{O(k)}$.

▶ **Claim 23.** *For all $n \in \mathbb{N}$, there exists a monotone-projection reduction* $\mathrm{SUB}_{G,n} \leq_{\mathrm{mp}} \mathrm{MODEL}_{\Phi,mn}$.

In order to define this monotone-projection reduction, let us identify $[mn]$ with the set $V(G^{\uparrow n})$ $(= V(G) \times [n])$. Variables X_e of $\mathrm{SUB}_{G,n}$ are indexed by potential edges $e \in E(G^{\uparrow n})$ in a subgraph $X \subseteq G^{\uparrow n}$. Variables Y_i of $\mathrm{MODEL}_{\Phi,mn}$ are indexed by the set

$$I := \big\{(R,(v,a),(w,b)) : R^{(2)} \in \sigma,\ (v,a),(w,b) \in V(G^{\uparrow n})\big\}.$$

(That is, I is the set of potential 2-tuples of relations of structures with universe $V(G^{\uparrow n})$.) Define the monotone projection $\rho : \{Y_i\}_{i\in I} \to \{X_e\}_{e\in E(G^{\uparrow n})} \cup \{0,1\}$ by

$$\rho : Y_{(R,(v,a),(w,b))} \mapsto \begin{cases} X_{\{(v,a),(w,b)\}} & \text{if } (v,w) \in R^{\mathcal{M}} \text{ and } v \neq w, \\ 1 & \text{if } (v,w) \in R^{\mathcal{M}} \text{ and } v = w, \\ 0 & \text{otherwise.} \end{cases}$$

We must show that the corresponding map

$$\rho^* : \{\text{subgraphs of } G^{\uparrow n}\} \to \{\text{structures with universe } V(G^{\uparrow n})\}$$

is in fact a reduction from $\mathrm{SUB}_{G,n}$ to $\mathrm{MODEL}_{\Phi,mn}$. That is, we must show that for any $X \subseteq G^{\uparrow n}$,

$$\mathrm{SUB}_{G,n}(X) = 1 \iff \mathrm{MODEL}_{\Phi,mn}(\rho^*(X)) = 1. \tag{3}$$

For the $\Longrightarrow$ direction of (3): Assume $\mathrm{SUB}_{G,n}(X) = 1$. Then $G^{(\alpha)} \subseteq X$ for some $\alpha \in [n]^{V(G)}$. The definition of ρ ensures that the map $v \mapsto (v,\alpha_v)$ is a homomorphism from $\mathcal{M}$ to the structure $\rho^*(X)$.[7] Since $\mathscr{C}$ is hom-preserved, it follows that $\rho^*(X) \in \mathscr{C}$ and therefore $\mathrm{MODEL}_{\Phi,mn}(\rho^*(X)) = 1$.

For the $\Longleftarrow$ direction of (3): Assume $\mathrm{MODEL}_{\Phi,mn}(\rho^*(X)) = 1$, that is, $\rho^*(X) \in \mathscr{C}$. By Lemma 22(a) there exist $\mathcal{N} \in \mathrm{MinCores}(\mathscr{C})$ and a homomorphism $\gamma : \mathcal{N} \to \rho^*(X)$. The definition of ρ ensures that the map $\pi : (v,i) \mapsto v$ is a homomorphism from $\rho^*(X)$ to $\mathcal{M}$.[8] The composition $\pi \circ \gamma$ is a homomorphism from $\mathcal{N}$ to $\mathcal{M}$. By Lemma 22(c), it is an isomorphism. Therefore, without loss of generality, we may assume that $\mathcal{M} = \mathcal{N}$ and $\pi \circ \gamma$ is the identity map on the universe $V(G)$ of $\mathcal{M}$. This means that $\pi(v) \in \{(v,a) : a \in [n]\}$ for all $v \in V(G)$. We may now define $\alpha \in [n]^{V(G)}$ as the unique element such that $\gamma : v \mapsto (v,\alpha_v)$ for all $v \in V(G)$. From the definition of ρ and the fact that $G = \mathrm{Gaif}(\mathcal{M})$, we infer that $G^{(\alpha)} \subseteq X$.[9] We conclude that $\mathrm{SUB}_{G,n}(X) = 1$, finishing the proof of Claim 23.

---

[7] To see why, suppose we have $(v,w) \in R^{\mathcal{M}}$ for some $R^{(2)} \in \sigma$. We must show that $((v,\alpha_v),(w,\alpha_w)) \in R^{\rho^*(X)}$. First, consider the case that $v \neq w$. The assumption $G^{(\alpha)} \subseteq X$ implies that $\{(v,\alpha_v),(w,\alpha_w)\} \in E(X)$. Since ρ maps the variable $Y_{(R,(v,\alpha_v),(w,\alpha_w))}$ to the variable $X_{\{(v,\alpha_v),(w,\alpha_w)\}}$ (which has value 1 for X), it follows that $((v,\alpha_v),(w,\alpha_w)) \in R^{\rho^*(X)}$. Finally, consider the case that $v = w$. In this case, ρ maps the variable $Y_{(R,(v,\alpha_v),(w,\alpha_w))}$ to the constant 1. So again we have $((v,\alpha_v),(w,\alpha_w)) \in R^{\rho^*(X)}$.

[8] In fact, this holds for every $X \subseteq G^{\uparrow n}$ independent of the assumption that $\mathrm{MODEL}_{\Phi,mn}(\rho^*(X)) = 1$. This follows from the observation that π is (in particular) a homomorphism from $\rho^*(G^{\uparrow n})$ to $\mathcal{M}$. To see why, consider any $((v,a),(w,b)) \in R^{\rho^*(G^{\uparrow n})}$ (corresponding to $Y_{(R,(v,a),(w,b))} = 1$). It must be the case that $(v,w) \in R^{\mathcal{M}}$, since the contrary assumption $(v,w) \notin R^{\mathcal{M}}$ would mean that ρ maps the variable $Y_{(R,(v,a),(w,b))}$ to 0.

[9] Consider an edge $\{(v,\alpha_v),(w,\alpha_v)\} \in E(G^{(\alpha)})$. By definition of $G^{(\alpha)}$, we have $\{v,w\} \in E(G)$. Since $G = \mathrm{Gaif}(\mathcal{M})$, there exists a relation $R^{(2)} \in \sigma$ such that $(v,w) \in R^{\mathcal{M}}$ or $(w,v) \in R^{\mathcal{M}}$. Without loss of generality, assume $(v,w) \in R^{\mathcal{M}}$. Since $\gamma : \mathcal{M} \to \rho^*(X)$ is a homomorphism, we have $(\gamma(v),\gamma(w)) = ((v,\alpha_v),(w,\alpha_w)) \in R^{\rho^*(X)}$. Since $(v,w) \in R^{\mathcal{M}}$ and $v \neq w$, the monotone projection ρ maps $Y_{(R,(v,\alpha_v),(w,\alpha_w))}$ to $X_{\{(v,\alpha_v),(w,\alpha_w)\}}$. It follows that $\{(v,\alpha_v),(w,\alpha_w)\} \in E(X)$. Therefore, $G^{(\alpha)}$ is a subgraph of X.

We proceed to show that $\mathbf{td}(G) \leq 2^{O(k)}$. By Corollary 11, we have $\mathrm{SUB}_{P_{\mathbf{td}(G)},n} \leq_{\mathrm{mp}} \mathrm{SUB}_{G,n}$. By Claim 23 and transitivity of $\leq_{\mathrm{mp}}$, it follows that $\mathrm{SUB}_{P_{\mathbf{td}(G)},n} \leq_{\mathrm{mp}} \mathrm{MODEL}_{\Phi,kn}$. Therefore, $\mu(\mathrm{SUB}_{P_{\mathbf{td}(G)},n}) \leq \mu(\mathrm{MODEL}_{\Phi,kn})$ for every standard complexity measure $\mu : \{\text{Boolean functions}\} \to \mathbb{N}$ (in particular, depth-k formula size). By Lemma 18 (the simulation of first-order logic by AC^0), there exist depth-k formulas of size $O((mn)^k)$ which compute $\mathrm{MODEL}_{\Phi,mn}$. Therefore, there exist depth-k formulas of size $O((mn)^k)$ which compute $\mathrm{SUB}_{P_{\mathbf{td}(G)},n}$. On the other hand, by Theorem 14, the depth-k formula size of $\mathrm{SUB}_{P_{\mathbf{td}(G)},n}$ is $n^{\Omega(\log \mathbf{td}(G))}$ for all sufficiently large n such that $k < \log\log n$. Therefore, we have $n^{\Omega(\log \mathbf{td}(G))} \leq O((mn)^k)$ for all sufficiently large n. Since m $(= |V(G)|)$ is constant, it follows that $k \geq \Omega(\log \mathbf{td}(G))$, that is, $\mathbf{td}(G) \leq 2^{O(k)}$. This completes the proof that $\beta(k) \leq 2^{O(k)}$ for binary signatures σ.

▶ **Remark 24.** In this argument, as an alternative to *depth-k formula size*, we may instead consider *fan-in $O(n)$ depth* (i.e. *fan-in cn depth* for a sufficiently large constant c) and appeal to Corollary 15 instead of Theorem 14.

Finally, we explain how to adapt the above argument when σ is an arbitrary finite relational signature. Here the variables of $\mathrm{MODEL}_{\Phi,kn}$ are indexed by the set

$$\big\{(R,(v_1,a_1),\dots,(v_t,a_t)) : R^{(t)} \in \sigma,\ (v_1,a_1),\dots,(v_t,a_t) \in V(G)\times[n]\big\}$$

and the reduction $\{\text{subgraphs of } G^{\uparrow n}\} \to \{\text{structures with universe } V(G^{\uparrow n})\}$ is defined by

$$Y_{(R,(v_1,a_1),\dots,(v_t,a_t))} \mapsto \begin{cases} \bigwedge_{1\leq i<j\leq t\,:\,v_i\neq v_j} X_{\{(v_i,a_i),(v_j,a_j)\}} & \text{if } (v_1,\dots,v_t)\in R^{\mathcal{M}},\\ 0 & \text{otherwise.}\end{cases}$$

(By convention, $\bigwedge_{i\in\emptyset} X_i = 1$.) Note that this reduction is not a monotone projection, as we are mapping each Y-variable to a conjunction of X-variables. This reduction is, however, computed by a single layer of constant fan-in AND gates. Therefore, under this reduction, any Boolean formula computing $\mathrm{MODEL}_{\Phi,kn}$ is converted to a Boolean formula computing $\mathrm{SUB}_{G,n}$ with an increase of 1 in depth and a constant factor increase in size. Other than this change, the rest of the argument is identical to the case of binary signatures.

7.2 Improvement to $\beta(k) \leq k^{O(1)}$

The upper bound $\beta(k) \leq 2^{O(k)}$ in the previous section relies on the *exponential approximation* of tree-depth in terms of the longest path, that is, $\log(\mathbf{lp}(G)+1) \leq \mathbf{td}(G) \leq \mathbf{lp}(G)$ (inequality (2)). To achieve a polynomial upper bound on $\beta(k)$, we require a *polynomial approximation* of tree-depth in terms of a few manageable classes "excluded minors". This realization led to a conjecture of the author, which was soon proved in joint work with Ken-ichi Kawarabayashi [25].

▶ **Theorem 25.** *Every graph G of tree-depth k satisfies one (or more) of the following conditions for $\ell = \widetilde{\Omega}(k^{1/5})$:*

(i) $\mathbf{tw}(G) \geq \ell$,
(ii) *G contains a path of length 2^ℓ, or*
(iii) *G contains a B_ℓ-minor.*

This result is analogous to the Polynomial Grid-Minor Theorem [12], which can be used to replace condition (i) with the condition that G contains an $\Omega(k^\varepsilon)\times\Omega(k^\varepsilon)$ grid minor for an absolute constant $\varepsilon > 0$. In cases (i) and (ii), Theorems 13 and 14 respectively imply

that SUB_G has AC^0 formula size $n^{\widetilde{\Omega}(\mathbf{td}(G)^{1/5})}$. This leaves only case (iii), where forthcoming work of the author [42] shows the following (via a generalization of the "pathset complexity" framework of [41]).

▶ **Theorem 26.** *The* AC^0 *formula size of* SUB_{B_k} *is* $n^{\Omega(k^\varepsilon)}$ *for an absolute constant* $\varepsilon > 0$.

Together Theorems 25 and 26 imply:

▶ **Theorem 27.** *For all graphs* G*, the* AC^0 *formula complexity of* SUB_G *is* $n^{\Omega(\mathbf{td}(G)^\varepsilon)}$ *for an absolute constant* $\varepsilon > 0$.

Plugging Theorem 27 into the argument in the previous subsection directly yields the polynomial upper bound $\beta(k) \leq k^{O(1)}$ of Theorem 4. (In fact, we get $\beta(k) \leq k^{1/\varepsilon}$ for the constant $\varepsilon > 0$ of Theorem 27.)

8 Comparison with the Method in (R. 2008)

In this section, for the sake of comparison, we summarize the model-theoretic approach of the original proof of Theorem 4 in [39]. The starting point in [39] is a new compactness-free proof of the classical Homomorphism Preservation Theorem, which moreover yields the stronger "equi-rank" version (Theorem 5). The proof is based on an operation mapping each structure $\mathcal{A}$ to an infinite co-retract $\Gamma(\mathcal{A})$. (We drop the assumption of the last two sections that structures are finite by default.) In order to state the key property of this operation, we introduce notation $\mathcal{A} \equiv_{\mathrm{FO}(k)} \mathcal{B}$ (resp. $\mathcal{A} \equiv_{\exists^+\mathrm{FO}(k)} \mathcal{B}$) denoting the statement that $\mathcal{A}$ and $\mathcal{B}$ satisfy the same first-order sentences (resp. existential-positive sentences) of quantifier-rank k.

▶ **Theorem 28.** *There is an operation* $\Gamma : \{\textit{structures}\} \to \{\textit{structures}\}$ *associating every structure* $\mathcal{A}$ *with a co-retract* $\Gamma(\mathcal{A}) \overset{\supseteq}{\to} \mathcal{A}$ *such that, for all structures* $\mathcal{A}$ *and* $\mathcal{B}$ *and* $k \in \mathbb{N}$*,*

$$\mathcal{A} \equiv_{\exists^+\mathrm{FO}(k)} \mathcal{B} \implies \Gamma(\mathcal{A}) \equiv_{\mathrm{FO}(k)} \Gamma(\mathcal{B}).$$

There is a straightforward proof that Theorem 28 implies Theorem 5 (see [39]). The structure $\Gamma(\mathcal{A})$ is the Fraïsse limit of the class of co-finite co-retracts of $\mathcal{A}$ (that is, structures $\mathcal{A}'$ such that $\mathcal{A}' \overset{\supseteq}{\to} \mathcal{A}$ and $A' \setminus A$ is finite). We remark that $\Gamma(\mathcal{A})$ is infinite, even when $\mathcal{A}$ is finite. For this reason, Theorem 28 says nothing in the setting of finite structures.

The Homomorphism Preservation Theorem on Finite Structures (Theorem 4) is proved in [39] by considering a sequence of finitary "approximations" of $\Gamma(\mathcal{A})$. (This is somewhat analogous to sense in which large random graph $G(n, 1/2)$ "approximate" the infinite Rado graph.)

▶ **Theorem 29.** *There is a computable function* $\beta : \mathbb{N} \to \mathbb{N}$ *and a sequence* $\{\Gamma_k\}_{k\in\mathbb{N}}$ *of operations* $\Gamma_k : \{\textit{finite structures}\} \to \{\textit{finite structures}\}$ *associating every finite structure* $\mathcal{A}$ *with a sequence* $\{\Gamma_k(\mathcal{A})\}_{k\in\mathbb{N}}$ *of finite co-retracts* $\Gamma_k(\mathcal{A}) \overset{\supseteq}{\to} \mathcal{A}$ *such that, for all finite structures* $\mathcal{A}$ *and* $\mathcal{B}$ *and* $k \in \mathbb{N}$*,*

$$\mathcal{A} \equiv_{\exists^+\mathrm{FO}(\beta(k))} \mathcal{B} \implies \Gamma_k(\mathcal{A}) \equiv_{\mathrm{FO}(k)} \Gamma_k(\mathcal{B}).$$

Theorem 4 follows directly from Theorem 29, inheriting the same quantifier-rank blow-up $\beta(k)$. The proof of Theorem 29 in [39] implies a non-elementary upper bound on $\beta(k)$. While the present paper improves the upper bound $\beta(k) \leq k^{O(1)}$ in Theorem 4, we remark that this it does not imply any improvement to $\beta(k)$ in Theorem 29.

9 Syntax vs. Semantics in Circuit Complexity

We conclude this paper by stating some consequences of our results in circuit complexity. Let HomPreserved denote the class of all homomorphism-preserved graph properties (for example, $\{G : \text{girth}(G) \leq 20 \text{ or clique-number}(G) \geq 10\}$). This is a semantic class, akin to the class Monotone of all monotone languages. The new proof in this paper of the Homomorphism Preservation Theorem on Finite Structures using AC^0 lower bounds is easily to imply the following "Homomorphism Preservation Theorem for (non-uniform) AC^0":

$$\text{HomPreserved} \cap \text{AC}^0 = \exists^+\text{FO} \quad (\subseteq \{\text{poly-size monotone DNFs}\}).$$

In other words, every homomorphism-preserved graph property in AC^0 is definable (among finite graphs) by an existential-positive first-order sentence and, therefore, also by a polynomial-size monotone DNF (moreover, with constant bottom fan-in). As a consequence, for every integer $d \geq 2$, we get a collapse of the AC^0 depth hierarchy with respect to homomorphism-preserved properties:

$$\text{HomPreserved} \cap \text{AC}^0[\text{depth } d] = \text{HomPreserved} \cap \text{AC}^0[\text{depth } d+1].$$

In contrast, it is known that $\text{AC}^0[\text{depth } d] \neq \text{AC}^0[\text{depth } d+1]$ by the Depth Hierarchy Theorem [21].

These results have an opposite nature to the "syntactic monotonicity $\neq$ semantic monotonicity" counterexamples of Ajtai and Gurevich [1] and Razborov [35] (as well as Tardos [45]), which respectively show that

$$\text{Monotone} \cap \text{AC}^0 \neq \text{Monotone-AC}^0 \quad \text{and} \quad \text{Monotone} \cap \text{P} \neq \text{Monotone-P}.$$

In light of the results of this paper, I feel that questions of syntax vs. semantics in circuit complexity are worth re-examining. For instance, so far as I know, there is no known separation between the *uniform average-case* monotone vs. non-monotone complexity of any monotone function in any well-studied class of Boolean circuits (AC^0, NC^1, etc.) It is plausible that syntactic monotonicity = semantic monotonicity in the average-case. Evidence for this viewpoint comes from the considering the *slice distribution* (that is, the uniform distribution on inputs of Hamming weight exactly $\lfloor n/2 \rfloor$). With respect to the slice distribution, it is known that monotone and non-monotone complexity are equivalent within a $\text{poly}(n)$ factor by a classic result of Berkowitz [11].

As for an even stronger "Homomorphism Preservation Theorem" in circuit complexity, we can state the following: if for every k, SUB_{P_k} requires unbounded-depth formula size $n^{\Omega(\log k)}$ (which is widely conjectured to be true) or even $n^{\omega_k \to \infty(1)}$, then $\text{HomPreserved} \cap \text{NC}^1 = \exists^+\text{FO}$. Therefore, I strongly believe in a "Homomorphism Preservation Theorem for NC^1". On the other hand, the homomorphism-preserved property of being 2-colorable a.k.a. non-bipartite ($= \{G : C_k \to G \text{ for any odd } k\}$) is in Logspace (this follows from Reingold's theorem [36]), yet it is not $\exists^+\text{FO}$-definable. Therefore, we may assert that $\text{HomPreserved} \cap \text{Logspace} \neq \exists^+\text{FO}$.

References

1 M. Ajtai and Y. Gurevich. Monotone versus positive. *J. ACM*, 34:1004–1015, 1987.

2 Miklos Ajtai. First-order definability on finite structures. *Annals of Pure and Applied Logic*, 45(3):211–225, 1989.

3 N. Alechina and Y. Gurevich. Syntax vs. semantics on finite structures. In J. Mychielski, G. Rozenberg, and A. Salomaa, editors, *Structures in Logic and Computer Science*, pages 14–33. Springer-Verlag, 1997.

4 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *Journal of the ACM*, 42(4):844–856, 1995.

5 Kazuyuki Amano. k-Subgraph isomorphism on AC^0 circuits. *Computational Complexity*, 19(2):183–210, 2010.

6 A. Atserias, A. Dawar, and Ph.G. Kolaitis. On preservation under homomorphisms and unions of conjunctive queries. *J. ACM*, 53(2):208–237, 2006.

7 Albert Atserias. On digraph coloring problems and treewidth duality. *European Journal of Combinatorics*, 29(4):796–820, 2008.

8 Albert Atserias, Anuj Dawar, and Martin Grohe. Preservation under extensions on well-behaved finite structures. *SIAM Journal on Computing*, 38(4):1364–1381, 2008.

9 Paul Beame, Russell Impagliazzo, and Toniann Pitassi. Improved depth lower bounds for small distance connectivity. *Computational Complexity*, 7(4):325–345, 1998.

10 Stephen Bellantoni, Toniann Pitassi, and Alasdair Urquhart. Approximation and small-depth frege proofs. *SIAM Journal on Computing*, 21(6):1161–1179, 1992.

11 S Berkowitz. On some relationships between monotone and nonmonotone circuit complexity. Technical report, Technical report, Department of Computer Science, University of Toronto, Canada, Toronto, Canada, 1982.

12 Chandra Chekuri and Julia Chuzhoy. Polynomial bounds for the grid-minor theorem. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 60–69. ACM, 2014.

13 Xi Chen, Igor C Oliveira, Rocco A Servedio, and Li-Yang Tan. Near-optimal small-depth lower bounds for small distance connectivity. *arXiv preprint arXiv:1509.07476*, 2015.

14 Anuj Dawar. Homomorphism preservation on quasi-wide classes. *Journal of Computer and System Sciences*, 76(5):324–332, 2010.

15 H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, 1996.

16 T. Feder and M.Y. Vardi. Homomorphism closed vs. existential positive. In *Proceedings of the 18th IEEE Symposium on Logic in Computer Science*, pages 310–320, 2003.

17 E. Grädel, P.G. Kolaitis, L. Libkin, M. Marx, J. Spencer, M.Y. Vardi, Y. Venema, and S. Weinstein. *Finite Model Theory and its Applications*. Springer, 2007.

18 E. Grädel and E. Rosen. On preservation theorems for two-variable logic. *Math. Logic Quart.*, 45:315–325, 1999.

19 Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM*, 54(1):1–24, 2007.

20 Y. Gurevich. Toward logic tailored for computational complexity. In M. M. Richter et al., editor, *Computation and Proof Theory*, pages 175–216. Springer Lecture Notes in Mathematics, 1984.

21 Johan Håstad. Almost optimal lower bounds for small depth circuits. In *STOC '86: Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, pages 6–20, 1986.

22 P. Hell and J. Nešetřil. The core of a graph. *Discrete Math.*, 109:117–126, 1992.

23 P. Hell and J. Nešetřil. *Graphs and Homomorphisms*. Oxford University Press, 2004.

24 W. Hodges. *Model Theory*. Cambridge University Press, 1993.

25 Ken ichi Kawarabayashi and Benjamin Rossman. An excluded-minor approximation of tree-depth. manuscript, 2016.

26 N. Immerman. *Descriptive Complexity Theory*. Graduate Texts in Computer Science. Springer, New York, 1999.

27 Yuan Li, Alexander Razborov, and Benjamin Rossman. On the ac0 complexity of subgraph isomorphism. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 344–353. IEEE, 2014.

28 L. Libkin. *Elements of Finite Model Theory*. Springer-Verlag, 2004.

29 R.C. Lyndon. Properties preserved under homomorphism. *Pacific J. Math.*, 9:129–142, 1959.

30 Dániel Marx. Can you beat treewidth? *Theory of Computing*, 6:85–112, 2010.

31 Jaroslav Nesetril and Patrice Ossona De Mendez. On first-order definable colorings. https://arxiv.org/abs/1403.1995, 2014.

32 J. Nešetřil and P. Ossona de Mendez. Tree depth, subgraph coloring and homomorphism bounds. *European J. Combin.*, 27(6):1022–1041, 2006.

33 J. Nešetřil and P. Ossona de Mendez. Sparsity (graphs, structures, and algorithms), algorithms and combinatorics, vol. 28, 2012.

34 Jürgen Plehn and Bernd Voigt. Finding minimally weighted subgraphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 18–29. Springer, 1990.

35 Alexander A Razborov. Lower bounds on the monotone complexity of some boolean functions. In *Dokl. Akad. Nauk SSSR*, volume 281, pages 798–801, 1985.

36 Omer Reingold. Undirected connectivity in log-space. *Journal of the ACM (JACM)*, 55(4):17, 2008.

37 E. Rosen. *Finite model theory and finite variable logic*. PhD thesis, University of Pennsylvania, 1995.

38 E. Rosen and S. Weinstein. Preservation theorems in finite model theory. In D. Leivant, editor, *Logic and Computational Complexity*, pages 480–502. Spring-Verlag, 1995.

39 Benjamin Rossman. Homomorphism preservation theorems. *Journal of the ACM (JACM)*, 55(3):15, 2008.

40 Benjamin Rossman. On the constant-depth complexity of k-clique. In *40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 721–730, 2008.

41 Benjamin Rossman. Formulas vs. circuits for small distance connectivity. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 203–212. ACM, 2014.

42 Benjamin Rossman. Lower bounds for subgraph isomorphism. manuscript, 2016.

43 A. Stolboushkin. Finite monotone properties. In *Proceedings of the 10th IEEE Symposium on Logic in Computer Science*, pages 324–330, 1995.

44 W. Tait. A counterexample to a conjecture of Scott and Suppes. *J. Symbolic Logic*, 24:15–16, 1959.

45 Éva Tardos. The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8(1):141–142, 1988.

46 Ryan Williams. Faster decision of first-order graph properties. In *Proc. 29th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 2014.

Low-Sensitivity Functions from Unambiguous Certificates

Shalev Ben-David[*1], **Pooya Hatami**[†2], **and Avishay Tal**[‡3]

1 **MIT, Cambridge, USA**
shalev@mit.edu
2 **DIMACS, Piscataway and IAS, Princeton, USA**
pooyahat@math.ias.edu
3 **IAS, Princeton, USA**
avishay.tal@gmail.com

Abstract

We provide new query complexity separations against sensitivity for total Boolean functions: a power 3 separation between deterministic (and even randomized or quantum) query complexity and sensitivity, and a power 2.22 separation between certificate complexity and sensitivity. We get these separations by using a new connection between sensitivity and a seemingly unrelated measure called one-sided unambiguous certificate complexity ($\mathrm{UC}_{\min}$). We also show that $\mathrm{UC}_{\min}$ is lower-bounded by fractional block sensitivity, which means we cannot use these techniques to get a super-quadratic separation between $\mathrm{bs}(f)$ and $\mathrm{s}(f)$.

Along the way, we give a power 1.22 separation between certificate complexity and one-sided unambiguous certificate complexity, improving the power 1.128 separation due to Göös [20]. As a consequence, we obtain an improved $\Omega(\log^{1.22} n)$ lower-bound on the co-nondeterministic communication complexity of the Clique vs. Independent Set problem.

1998 ACM Subject Classification F.1.3 Computation by Abstract Devices, Complexity Measures and Classes

Keywords and phrases Boolean functions, decision tree complexity, query complexity, sensitivity conjecture

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.28

1 Introduction

Sensitivity is one of the simplest complexity measures of a Boolean function. For $f : \{0,1\}^n \to \{0,1\}$ and $x \in \{0,1\}^n$, the sensitivity of x is the number of bits of x that, when flipped, change the value of $f(x)$. The sensitivity of f, denoted $\mathrm{s}(f)$, is the maximum sensitivity of any input x to f. Sensitivity lower bounds other important measures in query complexity, such as deterministic query complexity $\mathrm{D}(f)$, randomized query complexity $\mathrm{R}(f)$, certificate complexity $\mathrm{C}(f)$, and block sensitivity $\mathrm{bs}(f)$ (see Section 2 for definitions). $\sqrt{\mathrm{s}(f)}$ is a lower bound on quantum query complexity $\mathrm{Q}(f)$.

Despite its simplicity, sensitivity has remained mysterious. The other measures are polynomially related to each other: we have $\mathrm{bs}(f) \le \mathrm{C}(f) \le \mathrm{D}(f) \le \mathrm{bs}(f)^3$ and $\mathrm{Q}(f) \le$

* Partially supported by NSF.
† Partially supported by the National Science Foundation under agreement No. CCF-1412958.
‡ Supported by the Simons Collaboration on Algorithms and Geometry, and by the National Science Foundation grant No. CCF-1412958.

© S. Ben-David, P. Hatami, and A. Tal;
licensed under Creative Commons License CC-BY

$\mathrm{R}(f) \leq \mathrm{D}(f) \leq \mathrm{Q}(f)^6$. In contrast, no polynomial relationship connecting sensitivity to these measures is known, despite much interest (this problem was first posed by [31]. For a survey, see [26]. For recent progress, see [8, 13, 6, 4, 7, 9, 18, 39, 24, 25, 41]).

Until recently, the best known separation between sensitivity and any of these other measures was quadratic. Tal [41] showed a power 2.11 separation between $\mathrm{D}(f)$ and $\mathrm{s}(f)$. In this work, we improve this to a power 3 separation, and also show functions for which $\mathrm{Q}(f) = \tilde{\Omega}(\mathrm{s}(f)^3)$ and $\mathrm{C}(f) = \tilde{\Omega}(\mathrm{s}(f)^{2.22})$.

We do this by exploiting a new connection between sensitivity and a measure called one-sided unambiguous certificate complexity, which we denote by $\mathrm{UC}_{\min}(f)$. This measure, and particularly its two-sided version $\mathrm{UC}(f)$ (which is sometimes called subcube complexity), has received significant attention in previous work (e.g. [14, 17, 37, 11, 27, 23, 20, 21, 16, 5]), in part because it corresponds to partition number in communication complexity. Intuitively, $\mathrm{UC}_{\min}(f)$ is similar to (one-sided) certificate complexity, except that the certificates are required to be *unambiguous*: each input must be consistent with only one certificate. For a formal definition, see Section 2.5.

We prove the following theorem.

▶ **Theorem 1.** *For any $\alpha \in \mathbb{R}^+$, if there is a family of functions with $\mathrm{D}(f) = \tilde{\Omega}(\mathrm{UC}_{\min}(f)^{1+\alpha})$, then there is a family of functions with $\mathrm{D}(f) = \tilde{\Omega}(\mathrm{s}(f)^{2+\alpha})$. The same is true if we replace $\mathrm{D}(f)$ by $\mathrm{bs}(f), \mathrm{C}(f), \mathrm{R}(f), \mathrm{Q}(f)$, and many other measures.*

Theorem 1 can be generalized from sensitivity $\mathrm{s}(f)$ to bounded-size block sensitivity $\mathrm{bs}_{(k)}(f)$ (block sensitivity where each block is restricted to have size at most k). However, there is a constant factor loss that depends on k.

We observe that cheat sheet functions (as defined in [2]) have low $\mathrm{UC}_{\min}$; in particular, one of the functions in [2] already has a quadratic separation between $\mathrm{Q}(f)$ and $\mathrm{UC}_{\min}(f)$, giving a cubic separation between $\mathrm{Q}(f)$ and $\mathrm{s}(f)$.

▶ **Corollary 2.** *There is a family of functions with $\mathrm{Q}(f) = \tilde{\Omega}(\mathrm{s}(f)^3)$.*

To separate $\mathrm{C}(f)$ from $\mathrm{s}(f)$, we will use a function f with a significant gap between $\mathrm{C}(f)$ and $\mathrm{UC}_{\min}(f)$. Göös [20], as part of the proof of his exciting $\omega(\log n)$ lower-bound for communication complexity of clique versus independent set problem, gave a construction of a function f such that $\mathrm{C}(f) \geq \mathrm{UC}_{\min}(f)^\alpha$ for $\alpha \approx 1.128$. Using Göös's function [20] would give a family of functions with $\mathrm{C}(f) = \Omega(\mathrm{s}(f)^{2.128})$. We show that it is possible to obtain an even better separation (Theorem 4 below), leading to the following separation between $\mathrm{C}(f)$ and $\mathrm{s}(f)$.

▶ **Corollary 3.** *There is a family of functions with $\mathrm{C}(f) = \Omega(\mathrm{s}(f)^{2.22})$.*

New separation between C and $\mathrm{UC}_{\min}$

It is known that $\mathrm{C}(f) \leq \mathrm{UC}_{\min}(f)^2$ (e.g., [20]), and analogously in the communication complexity world $\mathbf{coNP}^{cc}(f) \leq \mathbf{UP}^{cc}(f)^2$ ([43]). Next, we discuss a polynomial separation between C and $\mathrm{UC}_{\min}$ due to [20] that uses function composition.

Throughout the years, Boolean function composition was used extensively to separate different complexity measures; a non-exhaustive list includes [1, 3, 12, 19, 28, 32, 33, 34, 42, 36, 38, 40, 41]. The natural idea is to exhibit some constant separation between any two measures: $M(f)$ and $N(f)$ (i.e., $M(f) < N(f)$ for a constant size function f) and then to prove that $M(f^k) \leq M(f)^k$ and $N(f^k) \geq N(f)^k$, for any $k \in \mathbb{N}$. This yields an infinite family of functions with polynomial separation between M and N, as $N(f^k) \geq M(f^k)^{\log N(f)/\log M(f)}$.

However, this approach does not work straightforwardly in an attempt to separate $\mathrm{UC}_{\min}$ from C, since it is not necessarily true that $\mathrm{UC}_{\min}(f^k) \leq \mathrm{UC}_{\min}(f)^k$. [20] overcomes this barrier by considering gadgets over a larger alphabet where the letters of the alphabet are weighted. He constructs such a gadget using projective planes, and further shows how to compose gadgets over a weighted alphabet in a way that behaves multiplicatively for both $\mathrm{UC}_{\min}$ and C. Finally, he shows how to simulate the weights and the larger alphabet with a Boolean function. The gadget f_k constructed by Göös satisfies $\mathrm{C}(f_k) = k^2 - k + 1$ and $\mathrm{UC}_{\min}(f_k) = \frac{k(k+1)}{2}$, whenever $k-1$ is a prime power. The optimum separation is obtained when $k = 8$, giving a separation exponent of $\log(57)/\log(36) \geq 1.128$.

Since $\mathrm{C}(f_k) \approx k^2$ and $\mathrm{UC}_{\min}(f_k) \approx k^2/2$ and the separation exponent is

$$\log(\mathrm{C}(f_k))/\log(\mathrm{UC}_{\min}(f_k)) \approx \log(k^2)/\log(k^2/2),$$

it seems that one should try to take k as small as possible. However, the additive terms affect smaller k's more significantly, making the optimum attained at $k = 8$. This motivated us to try and reduce the weights in other ways, in order to improve the exponent. To do so, we introduce *fractional weights*. The argument of Göös as is does not allow fractional weights, and in particular when Booleanizing the function, it seems inherent to use integer weights. We overcome this difficulty by considering fractional weights in intermediate steps of the construction, and then round them up at the end to get integral weights. We obtain the following separation.

▶ **Theorem 4** ($\mathrm{UC}_{\min}(f)$ vs $\mathrm{C}(f)$ - Improved)**.** *There exists an infinite family of Boolean functions* $f_n : \{0,1\}^n \to \{0,1\}$ *such that* $\mathrm{C}(f_n) \geq \tilde{\Omega}\left(\mathrm{UC}_{\min}(f_n)^{\frac{\log(38/3)}{\log(8)}}\right) \geq \Omega(\mathrm{UC}_{\min}(f_n)^{1.22})$.

Using the lifting theorem of Göös et al. [22] (see also [20, Appendix A]), Theorem 4 implies the following

▶ **Theorem 5** ($\mathbf{UP}^{cc}(f)$ vs $\mathbf{coNP}^{cc}(f)$)**.** *There exists an infinite family of Boolean functions* $f_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ *such that* $\mathbf{coNP}^{cc}(f_n) \geq \Omega(\mathbf{UP}^{cc}(f_n)^{1.22})$.

Hence, the exponent between $\mathbf{coNP}^{cc}$ and $\mathbf{UP}^{cc}$ is somewhere between 1.22 and 2. We conjecture the latter to be tight. Moreover, we get as a corollary an improved lower-bound for the conondeterministic communication complexity of the Clique vs Independent Set problem.

▶ **Corollary 6.** *There is a family of graphs* G *such that*

$$\mathbf{coNP}^{cc}(\mathrm{CIS}_G) \geq \Omega(\log^{1.22} n).$$

We refer the reader to [20] for a discussion on the Clique vs Independent Set problem that shows how Theorem 5 implies Corollary 6.

Limitations of Theorem 1

We note that $\mathrm{UC}_{\min}(f)$ upper bounds $\deg(f)$, so this technique cannot be used to get super-quadratic separations between $\deg(f)$ and $\mathrm{s}(f)$. A natural question is whether we can use Theorem 1 to get a super-quadratic separation between $\mathrm{bs}(f)$ and $\mathrm{s}(f)$. To do so, it would suffice to separate $\mathrm{bs}(f)$ from $\mathrm{UC}_{\min}(f)$. It would even suffice to separate randomized certificate complexity $\mathrm{RC}(f)$ (a measure larger than $\mathrm{bs}(f)$) from $\mathrm{UC}_{\min}(f)$, because of the following theorem.

▶ **Theorem 7** ([28, Corollary 3.2])**.** *If there exists a family of functions with* $\mathrm{RC}(f) \geq \Omega(\mathrm{s}(f)^{2+\alpha})$*, then there exists a family of functions with* $\mathrm{bs}(g) \geq \Omega(\mathrm{s}(g)^{2+\alpha-o(1)})$.

Unfortunately, we show that separating $\mathrm{RC}(f)$ from $\mathrm{UC}_{\min}(f)$ is impossible. We conclude that Theorem 1 cannot be used to super-quadratically separate $\mathrm{bs}(f)$ from $\mathrm{s}(f)$.

▶ **Theorem 8.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. Then $\mathrm{RC}(f) \leq 2\,\mathrm{UC}_{\min}(f) - 1$.*

We show that the factor of 2 in Theorem 8 is necessary. In Appendix A we strengthen this theorem to show that $\mathrm{RC}(f)$ also lower bounds one-sided conical junta degree.

Organization

In Section 2, we briefly define the many complexity measures mentioned here, and discuss the known relationships between them. In Section 3, we prove Theorem 1 and Corollary 2. In Section 4 we prove Theorem 4, from which Corollary 3 follows. In Section 5, we discuss a failed attempt to get a new separation between $\mathrm{bs}(f)$ and $\mathrm{s}(f)$, and in the process we prove Theorem 8.

2 Preliminaries

2.1 Query Complexity

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. Let A be a deterministic algorithm that computes $f(x)$ on input $x \in \{0,1\}^n$ by making queries to the bits of x. The worst-case number of queries A makes (over choices of x) is the query complexity of A. The minimum query complexity of any deterministic algorithm computing f is the deterministic query complexity of f, denoted by $\mathrm{D}(f)$.

We define the bounded-error randomized (respectively quantum) query complexity of f, denoted by $\mathrm{R}(f)$ (respectively $\mathrm{Q}(f)$), in an analogous way. We say an algorithm A computes f with bounded error if $\mathbf{Pr}[A(x) = f(x)] \geq 2/3$ for all $x \in \{0,1\}^n$, where the probability is over the internal randomness of A. Then $\mathrm{R}(f)$ (respectively $\mathrm{Q}(f)$) is the minimum number of queries required by any randomized (respectively quantum) algorithm that computes f with bounded error. It is clear that $\mathrm{Q}(f) \leq \mathrm{R}(f) \leq \mathrm{D}(f)$. For more details on these measures, see the survey by Buhrman and de Wolf [15].

2.2 Partial Assignments and Certificates

A partial assignment is a string $p \in \{0,1,*\}^n$ representing partial knowledge of a string $x \in \{0,1\}^n$. Two partial assignments are consistent if they agree on all entries where neither has a $*$. We will identify p with the set $\{(i, p_i) : p_i \neq *\}$. This allows us to write $p \subseteq x$ to denote that the string x is consistent with the partial assignment p. We observe that if p and q are consistent partial assignments, then $p \cup q$ is also a partial assignment. The size of a partial assignment p is $|p|$, the number of non-$*$ entries in p. The support of p is the set $\{i \in [n] : p_i \neq *\}$.

Fix a Boolean function $f : \{0,1\}^n \to \{0,1\}$. We say a partial assignment p is a certificate (with respect to f) if $f(x)$ is the same for all strings $x \supseteq p$. If $f(x) = 0$ for such strings, we say p is a 0-certificate; otherwise, we say p is a 1-certificate. We say p is a certificate for the string x if p is consistent with x. We use $\mathrm{C}_x(f)$ to denote the size of the smallest certificate for x. We then define the certificate complexity of f as $\mathrm{C}(f) := \max_{x \in \{0,1\}^n} \mathrm{C}_x(f)$. We also define the one-sided measures $\mathrm{C}_0(f) := \max_{x \in f^{-1}(0)} \mathrm{C}_x(f)$ and $\mathrm{C}_1(f) := \max_{x \in f^{-1}(1)} \mathrm{C}_x(f)$.

2.3 Sensitivity and Block Sensitivity

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function, and let $x \in \{0,1\}^n$ be a string. A block is a subset of $[n]$. If B is a block, we denote by x^B the string we get from x by flipping the bits in B; that is, $x_i^B = x_i$ if $i \notin B$, and $x^B = 1 - x_i$ if $i \in B$. For a bit i, we also use x^i to denote $x^{\{i\}}$.

We say that a block B is sensitive for x (with respect to f) if $f(x^B) \neq f(x)$. We say a bit i is sensitive for x if the block $\{i\}$ is sensitive for x. The maximum number of disjoint blocks that are all sensitive for x is called the block sensitivity of x (with respect to f), denoted by $\mathrm{bs}_x(f)$. The number of sensitive bits for x is called the sensitivity of x, denoted by $\mathrm{s}_x(f)$. Clearly, $\mathrm{bs}_x(f) \geq \mathrm{s}_x(f)$, since $\mathrm{s}_x(f)$ has the same definition as $\mathrm{bs}_x(f)$ except the size of the blocks is restricted to 1.

We now define the measures $\mathrm{s}(f)$, $\mathrm{s}_0(f)$, and $\mathrm{s}_1(f)$ analogously to $\mathrm{C}(f)$, $\mathrm{C}_0(f)$, and $\mathrm{C}_1(f)$. That is, $\mathrm{s}(f)$ is the maximum of $\mathrm{s}_x(f)$ over all x, $\mathrm{s}_0(f)$ is the maximum where x ranges over 0-inputs to f, and $\mathrm{s}_1(f)$ is the maximum over 1-inputs. We define $\mathrm{bs}(f)$, $\mathrm{bs}_0(f)$, and $\mathrm{bs}_1(f)$ similarly.

2.4 Fractional Block Sensitivity

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function, and let $x \in \{0,1\}^n$ be a string. Note that the support of any certificate p of x must have non-empty intersection with every sensitive block B of x; this is because otherwise, x^B would be consistent with p, which is a contradiction since $f(x^B) \neq f(x)$.

Note further that any subset S of $[n]$ that intersects with all sensitive blocks of x gives rise to a certificate x_S for x. This is because if x_S was not a certificate, there would be an input $y \supseteq x_S$ with $f(y) \neq f(x)$. If we write $y = x^B$, where B is the set of bits where x and y disagree, then B would be a sensitive block that is disjoint from S, which contradicts our assumption on S.

This means the certificate complexity $\mathrm{C}_x(f)$ of x is the hitting number for the set system of sensitive blocks of x (that is, the size of the minimum set that intersects all the sensitive blocks). Furthermore, the block sensitivity $\mathrm{bs}_x(f)$ of x is the packing number for the same set system (i.e. the maximum number of disjoint sets in the system). It is clear that the hitting number is always larger than the packing number, because if there are k disjoint sets we need at least k domain elements in order to have non-empty intersection with all the sets.

Moreover, we can define the fractional certificate complexity of x as the fractional hitting number of the set system; that is, the minimum amount of non-negative weight we can distribute among the domain elements $[n]$ so that every set in the system gets weight at least 1 (where the weight of a set is the sum of the weights of its elements). We can also define the fractional block sensitivity of x as the fractional packing number of the set system; that is, the maximum amount of non-negative weight we can distribute among the sets (blocks) so that every domain element gets weight at most 1 (where the weight of a domain element is the sum of the weights of the sets containing that element).

It is not hard to see that the fractional hitting and packing numbers are the solutions to dual linear programs, which means they are equal. We denote them by $\mathrm{RC}_x(f)$ for "randomized certificate complexity", following the original notation as introduced by Aaronson [1] (we warn that our definition differs by a constant factor from Aaronson's original definition). We define $\mathrm{RC}(f)$, $\mathrm{RC}_0(f)$, and $\mathrm{RC}_1(f)$ in the usual way. For more properties of $\mathrm{RC}(f)$, see [1] and [28].

2.5 Unambiguous Certificate Complexity

Fix $f : \{0,1\}^n \to \{0,1\}$. We call a set of partial assignments U an unambiguous collection of 0-certificates for f if

1. Each partial assignment in U is a 0-certificate (with respect to f)
2. For each $x \in f^{-1}(0)$, there is some $p \in U$ with $p \subseteq x$
3. No two partial assignments in U are consistent.

We then define $\mathrm{UC}_0(f)$ to be the minimum value of $\max_{p \in U} |p|$ over all choices of such collections U. We define $\mathrm{UC}_1(f)$ analogously, and set $\mathrm{UC}(f) := \max\{\mathrm{UC}_0(f), \mathrm{UC}_1(f)\}$. We also define the one-sided version, $\mathrm{UC}_{\min}(f) := \min\{\mathrm{UC}_0(f), \mathrm{UC}_1(f)\}$.

2.6 Degree Measures

A polynomial q in the variables $x_1, x_2, \ldots, x_n$ is said to represent the function $f : \{0,1\}^n \to \{0,1\}$ if $q(x) = f(x)$ for all $x \in \{0,1\}^n$. q is said to ϵ-approximate f if $q(x) \in [0, \epsilon]$ for all $x \in f^{-1}(0)$ and $q(x) \in [1-\epsilon, 1]$ for all $x \in f^{-1}(1)$. The degree of f, denoted by $\deg(f)$, is the minimum degree of a polynomial representing f. The ϵ-approximate degree, denoted by $\widetilde{\deg}^{\epsilon}(f)$, is the minimum degree of a polynomial ϵ-approximating f. We will omit ϵ when $\epsilon = 1/3$. [10] showed that $\mathrm{D}(f) \geq \deg(f)$, $\mathrm{R}(f) \geq \widetilde{\deg}(f)$, and $\mathrm{Q}(f) \geq \widetilde{\deg}(f)/2$.

We also define non-negative variants of degree. For each partial assignment p we identify a polynomial $p(x) := (\Pi_{i:\, p_i=1} x_i)\,(\Pi_{i:\, p_i=0}(1-x_i))$. We note that $p(x) = 1$ if $p \subseteq x$ and $p(x) = 0$ otherwise, and also that the degree of $p(x)$ is $|p|$. We say a polynomial is non-negative if it is of the form $\sum_p w_p p(x)$, where $w_p \in \mathbb{R}^+$ are non-negative weights. For such a sum, define its degree as $\max_{p:\, w_p > 0} |p|$. Define its average degree as the maximum over $x \in \{0,1\}^n$ of $\sum_{p:\, p \subseteq x} w_p |p|$. We note that if a non-negative polynomial q satisfies $|q(x)| \in [0,1]$ for all $x \in \{0,1\}^n$, then the average degree of q is at most its degree. Moreover, if all the monomials in q have the same size and $q(x) = 1$ for some $x \in \{0,1\}^n$, the degree and average degree of q are equal.

We define the non-negative degree of f as the minimum degree of a non-negative polynomial representing f. We note that this is a one-sided measure, since it may change when f is negated; we therefore denote it by $\deg_1^+(f)$, and use $\deg_0^+(f)$ for the degree of a non-negative polynomial representing the negation of f. We let $\deg^+(f)$ be the maximum of the two, and let $\deg^+_{\min}(f)$ be the minimum. We also define $\mathrm{avdeg}_1^+(f)$ as the minimum average degree of a non-negative polynomial representing f, with the other corresponding measures defined analogously. Finally, we define the approximate variants of these, denoted by (for example) $\widetilde{\deg}_1^{+,\epsilon}(f)$, in a similar way, except the polynomials need only to ϵ-approximate f.

2.7 Known Relationships

2.7.1 Two-Sided Measures

We describe some of the known relationships between these measures. To start with, we have

$$\mathrm{s}(f) \leq \mathrm{bs}(f) \leq \mathrm{RC}(f) \leq \mathrm{C}(f) \leq \mathrm{UC}(f) \leq \mathrm{D}(f),$$

where the last inequality holds because for each deterministic algorithm A, the partial assignments defined by the input bits A examines when run on some $x \in \{0,1\}^n$ form an unambiguous collection of certificates. We also have

$$\widetilde{\deg}(f) \leq 2\mathrm{Q}(f), \quad \widetilde{\deg}^{+}(f) \leq \mathrm{R}(f), \quad \deg^+(f) \leq \mathrm{D}(f),$$

with $\widetilde{\deg}(f) \leq \widetilde{\deg}^{+}(f) \leq \deg^{+}(f)$ and $\mathrm{Q}(f) \leq \mathrm{R}(f) \leq \mathrm{D}(f)$.

[10] showed $\mathrm{D}(f) \leq \mathrm{bs}(f)\,\mathrm{C}(f)$, and [31] showed $\mathrm{C}(f) \leq \mathrm{bs}(f)^2$. From this we conclude that $\mathrm{D}(f) \leq \mathrm{C}(f)^2$ and $\mathrm{D}(f) \leq \mathrm{bs}(f)^3$. [28] showed $\sqrt{\mathrm{RC}(f)} = O(\widetilde{\deg}(f))$; thus

$$\mathrm{D}(f) \leq \mathrm{bs}(f)^3 \leq \mathrm{RC}(f)^3 = O(\widetilde{\deg}(f)^6) = O(\mathrm{Q}(f)^6),$$

so the above measures are polynomially related (with the exception of sensitivity). Other known relationships are $\mathrm{RC}(f) = O(\mathrm{R}(f))$ (due to [1]), $\mathrm{D}(f) \leq \mathrm{bs}(f)\deg(f) \leq \deg(f)^3$ (due to [30]), and $\deg^{+}(f) \leq \mathrm{UC}(f)$ (since we can get a polynomial representing f by summing up the polynomials corresponding to unambiguous 1-certificates of f).

2.7.2 One-Sided Measures

One-sided measures such as $\mathrm{C}_1(f)$ are not polynomially related to the rest of the measures above, as can be seen from $\mathrm{C}_1(\mathrm{OR}_n) = 1$. This makes them less interesting to us. On the other hand, the one sided measures $\deg^{+}_{\min}(f)$, $\widetilde{\deg}^{+}_{\min}(f)$, and $\mathrm{UC}_{\min}(f)$ are polynomially related to the rest. An easy way to observe this is to note that $\widetilde{\deg}^{+}_{\min}(f) \geq \widetilde{\deg}(f)$, which follows from the fact that $\widetilde{\deg}(f) \leq \widetilde{\deg}^{+}_{1}(f)$ and that $\widetilde{\deg}(f)$ is invariant under negating f. Similarly, $\deg(f) \leq \deg^{+}_{\min}(f)$. We also have

$$\widetilde{\deg}^{+}_{\min}(f) \leq \deg^{+}_{\min}(f) \leq \mathrm{UC}_{\min}(f),$$

where the last inequality holds since we can form a non-negative polynomial representing f by summing up the polynomials corresponding to a set of unambiguous 1-certificates.

An additional useful inequality is $\mathrm{D}(f) \leq \mathrm{UC}_{\min}(f)^2$. The analogous statement in communication complexity was shown by [43]. The query complexity version of the proof can be found in [20].

3 Sensitivity and Unambiguous Certificates

We start by defining a transformation that takes a function f and modifies it so that $\mathrm{s}_0(f)$ decreases to 1. This transformation might cause $\mathrm{s}_1(f)$ to increase, but we will argue that it will remain upper bounded by $3\,\mathrm{UC}_1(f)$. We will also argue that other measures, such as $\mathrm{D}(f)$, do not decrease. This transformation is motivated by the construction of [41] that was used to give a power 2.115 separation between $\mathrm{D}(f)$ and $\mathrm{s}(f)$.

▶ **Definition 9** (Desensitizing Transformation). Let $f : \{0,1\}^n \to \{0,1\}$. Let U be an unambiguous collection of 1-certificates for f, each of size at most $\mathrm{UC}_1(f)$. For each $x \in f^{-1}(1)$, let $p_x \in U$ be the unique certificate in U consistent with x. The desensitized version of f is the function $f' : \{0,1\}^{3n} \to \{0,1\}$ defined by $f'(xyz) = 1$ if and only if $f(x) = f(y) = f(z) = 1$ and $p_x = p_y = p_z$.

The following lemma illustrates key properties of f'.

▶ **Lemma 10** (Desensitization). *Let f' be the desensitized version of $f : \{0,1\}^n \to \{0,1\}$. Then $\mathrm{s}_0(f') = 1$ and $\mathrm{UC}_1(f') \leq 3\,\mathrm{UC}_1(f)$. Also, for any complexity measure*

$$M \in \{\mathrm{D}, \mathrm{R}, \mathrm{Q}, \mathrm{C}, \mathrm{C}_0, \mathrm{C}_1, \mathrm{bs}, \mathrm{bs}_0, \mathrm{bs}_1, \mathrm{RC}, \mathrm{RC}_0, \mathrm{RC}_1, \mathrm{UC}, \mathrm{UC}_0, \mathrm{UC}_1, \mathrm{UC}_{\min}, \deg, \deg^{+}, \widetilde{\deg}, \widetilde{\deg}^{+}\},$$

we have $M(f') \geq M(f)$.

Proof. We start by upper bounding $\mathrm{s}_0(f')$. Consider any 0-input xyz to f' which has at least one sensitive bit. Pick a sensitive bit i of this input; without loss of generality, this bit is inside the x part of the input. Since flipping i changes xyz to a 1-input for f', we must have $f(x^i) = f(y) = f(z) = 1$ and $p_{x^i} = p_y = p_z$. In particular, it must hold that $f(y) = f(z) = 1$ and $p_y = p_z$. Let $p := p_y$, so $p = p_z$ and $p = p_{x^i}$. Since $f(xyz) = 0$, it must be the case that x is not consistent with p. Since p is consistent with x^i, it must be the case that p and x disagree exactly on the bit i.

Now, it's clear that xyz cannot have any sensitive bits inside the y part of the input, because then x would not be consistent with p_z. Similarly, xyz cannot have sensitive bits in the z part of the input. Any sensitive bits inside the x part of the input must make x consistent with p; but x disagrees with p on bit i, so this must be the only sensitive bit. It follows that the sensitivity of xyz is at most 1, as desired. We conclude that $\mathrm{s}_0(f') = 1$.

Next, we upper bound UC_1. Define $U' := \{ppp : p \in U\} \subseteq \{0,1,*\}^{3n}$. We show that this is an unambiguous collection of 1-certificates for f'. First, note that for $p \in U$, if $ppp \subseteq xyz$, then $f(x) = f(y) = f(z) = 1$ and $p_x = p_y = p_z = p$, so $f'(xyz) = 1$. Thus U' is a set of 1-certificates. Next, if xyz is a 1-input for f', then $f(x) = f(y) = f(z) = 1$ and $p_x = p_y = p_z$, which means $p_x p_x p_x \subseteq xyz$. Since $p_x \in U$, we have $p_x p_x p_x \in U'$. Finally, if $ppp, qqq \in U'$ with $ppp \neq qqq$, then $p \neq q$ and $p, q \in U$, which means p and q are inconsistent. This means ppp and qqq are inconsistent. This concludes the proof that U' is an unambiguous collection of 1-certificates for f'. We have $\max_{ppp \in U'} |ppp| = 3 \cdot \max_{p \in U} |p| = 3 \cdot \mathrm{UC}_1(f)$, so $\mathrm{UC}_1(f') \leq 3 \cdot \mathrm{UC}_1(f)$.

Finally, we show that almost all complexity measures do not decrease in the transition from f to f'. To see this, note that we can restrict f' to the promise that all inputs come from the set $\{xyz \in \{0,1\}^{3n} : x = y = z\}$. Under this promise, the function f' is simply the function f with each input bit occurring 3 times. But tripling input bits in this way does not affect the usual complexity measures (among the measures defined in Section 2, sensitivity is the only exception), and restricting to a promise can only cause them to decrease. This means that f' has higher complexity than f under almost any measure. ◀

We now prove Theorem 1, which we restate here for convenience.

▶ **Theorem 1.** *For any $\alpha \in \mathbb{R}^+$, if there is a family of functions with $\mathrm{D}(f) = \tilde{\Omega}(\mathrm{UC}_{\min}(f)^{1+\alpha})$, then there is a family of functions with $\mathrm{D}(f) = \tilde{\Omega}(\mathrm{s}(f)^{2+\alpha})$. The same is true if we replace $\mathrm{D}(f)$ by $\mathrm{bs}(f), \mathrm{C}(f), \mathrm{R}(f), \mathrm{Q}(f)$, and many other measures.*

Proof. Fix $f : \{0,1\}^n \to \{0,1\}$ from the family for which $\mathrm{D}(f) = \tilde{\Omega}(\mathrm{UC}_{\min}(f)^{1+\alpha})$. By negating f if necessary, assume $\mathrm{UC}_1(f) = \mathrm{UC}_{\min}(f)$. Apply the desensitizing transformation to get f'. By Lemma 10, we have $\mathrm{s}_0(f') \leq 1$ and $\mathrm{s}_1(f') \leq \mathrm{UC}_1(f') \leq 3\,\mathrm{UC}_{\min}(f)$, and also $\mathrm{D}(f') \geq \mathrm{D}(f)$. We now consider the function $\hat{f} := \mathrm{OR}_{3\,\mathrm{UC}_{\min}(f)} \circ f'$. It is not hard to see that $\mathrm{s}_0(\hat{f}) \leq 3\,\mathrm{UC}_{\min}(f)$ and $\mathrm{s}_1(\hat{f}) = \mathrm{s}_1(f') \leq 3\,\mathrm{UC}_{\min}(f)$, so $\mathrm{s}(\hat{f}) \leq 3\,\mathrm{UC}_{\min}(f)$.

We now analyze $\mathrm{D}(\hat{f})$. We have $\mathrm{D}(f') \geq \mathrm{D}(f)$; since deterministic query complexity satisfies a perfect composition theorem, we have

$$\mathrm{D}(\hat{f}) = \mathrm{D}(\mathrm{OR}_{3\,\mathrm{UC}_{\min}(f)})\,\mathrm{D}(f') \geq 3\,\mathrm{UC}_{\min}(f)\,\mathrm{D}(f) = \tilde{\Omega}(\mathrm{UC}_{\min}(f)^{2+\alpha}) = \tilde{\Omega}(\mathrm{s}(\hat{f})^{2+\alpha}).$$

This concludes the proof for deterministic query complexity.

For other measures, we need the following properties: first, that the measure is invariant under negating the function (so that we can assume $\mathrm{UC}_{\min}(f) = \mathrm{UC}_1(f)$ without loss of generality); second, that the measure satisfies a composition theorem, at least in the case that the outer function is OR; and finally, that the measure is large for the OR function. We

note that the measures C, bs, RC, R, and Q all satisfy a composition theorem of the form $M(\mathrm{OR} \circ g) \geq \Omega(M(\mathrm{OR})M(g))$; for the first three measures, this can be found in [19], for R it can be found in [21], and for Q it follows from a general composition theorem [35, 29]. Moreover, $\mathrm{bs}(\mathrm{OR}_n) = \mathrm{C}(\mathrm{OR}_n) = \mathrm{RC}(\mathrm{OR}_n) = n$ and $\mathrm{R}(\mathrm{OR}_n) = \Omega(n)$. This completes the proof for these measures; for Q, we will have to work harder, since $\mathrm{Q}(\mathrm{OR}_n) = \Theta(\sqrt{n})$.

For quantum query complexity, the trick will be to use the function "Block k-sum" defined in [2]. It has the property that all inputs have certificates that use very few 0 bits. Actually, we'll swap the 0s and 1s so that all inputs have certificates that use very few 1 bits. When $k = \log n$ (where n the size of the input), we denote this function by BSUM_n. [2] showed that $\mathrm{Q}(\mathrm{BSUM}_n) = \tilde{\Omega}(n)$, and every input has a certificate with $O(\log^3 n)$ ones.

Consider the function $\hat{f} := \mathrm{BSUM}_{\mathrm{UC}_{\min}(f)} \circ f'$. We have $\mathrm{Q}(\hat{f}) = \mathrm{Q}(\mathrm{BSUM}_{\mathrm{UC}_{\min}(f)})\mathrm{Q}(f') = \tilde{\Omega}(\mathrm{UC}_{\min}(f)\mathrm{Q}(f))$. We now analyze the sensitivity of $\hat{f}$. Fix an input z to $\hat{f} = \mathrm{BSUM}_{\mathrm{UC}_{\min}(f)} \circ f'$. This input consists of $\mathrm{UC}_{\min}(f)$ inputs to f', which, when evaluated, form an input y to $\mathrm{BSUM}_{\mathrm{UC}_{\min}(f)}$. Note that some of the inputs to f' correspond to sensitive bits of y (with respect to $\mathrm{BSUM}_{\mathrm{UC}_{\min}(f)}$); the sensitive bits of z are then simply the sensitive bits of those inputs. Now, consider the certificate of y that uses only $O(\log^3 \mathrm{UC}_{\min}(f))$ bits that are 1. Since it is a certificate, it must contain all the sensitive bits of y; thus at most $O(\log^3 \mathrm{UC}_{\min}(f))$ of the 1 bits of y are sensitive. It follows that the number of sensitive bits of z is at most $\mathrm{UC}_{\min}(f)\,\mathrm{s}_0(f') + O(\log^3 \mathrm{UC}_{\min}(f))\,\mathrm{s}_1(f') = \tilde{O}(\mathrm{UC}_{\min}(f))$. This concludes the proof. ◀

It is not hard to see that the same approach can yield separations against bounded-size block sensitivity (where the blocks are restricted to have size at most k). To do this, we need the desensitizing construction to repeat the inputs $2k+1$ times instead of 3 times. Instead of increasing to $3\,\mathrm{UC}_{\min}(f)$, the bounded-size block sensitivity would increase to $(2k+1)\,\mathrm{UC}_{\min}(f)$, and the deterministic query complexity would increase to $(2k+1)\,\mathrm{D}(f)$. When k is constant, we get the same asymptotic separations as for sensitivity.

We now construct separations against $\mathrm{UC}_{\min}$. This proves Corollary 2 and Corollary 3.

▶ **Corollary 2.** *There is a family of functions with* $\mathrm{Q}(f) = \tilde{\Omega}(\mathrm{s}(f)^3)$.

Proof. By Theorem 1, it suffices to construct a family of functions with $\mathrm{Q}(f) = \tilde{\Omega}(\mathrm{UC}_{\min}(f)^2)$. Our function will be a cheat sheet function $\mathrm{BKK}_{\mathrm{CS}}$ from [2] that quadratically separates quantum query complexity from exact degree. This function has quantum query complexity quadratically larger than $\mathrm{UC}_{\min}$, as shown in [5]. ◀

▶ **Corollary 3.** *There is a family of functions with* $\mathrm{C}(f) = \Omega(\mathrm{s}(f)^{2.22})$.

Proof. In Theorem 4, we construct a family of functions with $\mathrm{C}(f) = \tilde{\Omega}(\mathrm{UC}_{\min}(f)^{\frac{\log(38/3)}{\log(8)}})$. Thus, by Theorem 1, we can construct a family of functions with $\mathrm{C}(f) = \tilde{\Omega}(\mathrm{s}(f)^{1+\frac{\log(38/3)}{\log(8)}}) = \Omega(\mathrm{s}(f)^{2.22})$. ◀

4 Improved separation between UC_1 and C

In this section we prove Theorem 4, building on the proof by [20]. Our main contribution is to show how to adapt the argument in [20] to allow for fractional weights. We finally give a fractional weighting scheme that leads to our improved separation. We observe that in order to obtain our final result, one can just take Göös's construction and reweigh it in the end. Nonetheless, we include the full details here to show that any gadget with a separation between UC_1 and C implies an asymptotic separation (which was not explicit in [20]).

Throughout the section, Σ and Γ will denote finite sets that correspond to input and output alphabets of our functions. We shall assume that 0 is not in Σ, and will discuss functions $f : (\{0\} \cup \Sigma)^n \to \Gamma$ where 0 is a special symbol treated differently than others.

4.1 Certificates and Weighted-Certificates for Large-Alphabet Functions

We generalize the definition of certificates from Boolean functions to functions with arbitrary input and output alphabets.

▶ **Definition 11** (Multi-valued Certificates, Simple Certificates)**.** A certificate for a function $f : (\{0\} \cup \Sigma)^n \to \Gamma$ is a cartesian product of sets $S_1 \times S_2 \times \ldots S_n$ where each $S_i \subseteq \{0\} \cup \Sigma$ is a non-empty set and such that all $y \in S_1 \times S_2 \times \ldots S_n$ have the same f-value.

A simple certificate for f is a certificate where each S_i is either: (i) $\{0\} \cup \Sigma$, or (ii) S_i contains exactly one element, and this element is from Σ (i.e., not the 0 element).[1]

We define the size of a certificate as the number of i's such that $S_i \neq (\{0\} \cup \Sigma)$. For $x \in (\{0\} \cup \Sigma)^n$, we denote by $\mathrm{C}(f, x)$ the size of the smallest certificate for f which contains x.

For a set $T \subseteq \Gamma$ we say that $S_1 \times \ldots \times S_n$ certifies that "$f(\cdot) \in T$" if this is true for any $y \in S_1 \times \ldots \times S_n$. When $T = \Gamma \setminus \{i\}$ we write "$f(\cdot) \neq i$" for shorthand.

▶ **Definition 12** (Weight Schemes, Certificate Weights)**.** Let $w : \Sigma \to \mathbb{R}^+$ be a non-negative weight function. A weight scheme is a mapping, w, associating positive real numbers to non-empty subsets of $\{0\} \cup \Sigma$ such that:

1. If $S = \{i\}$, for some $i \in \Sigma$, then the weight of S is $w(i)$.
2. If $S = (\{0\} \cup \Sigma)$, then the weight of S is 0.
3. If $0 \in S$ an $S \neq (\{0\} \cup \Sigma)$, then the weight of S is $\max_{i \in \Sigma \setminus S}\{w(i)\}$. (In particular, if $S = \{0\}$ then the weight of S is $\max_{i \in \Sigma}\{w(i)\}$.)

(Note that we did not specify the weight of sets S of at least two elements which do not contain 0, as they will not be used in our analysis.)

The weight of a certificate $S_1 \times \ldots \times S_n$ is simply $\sum_{i=1}^{n} w(S_i)$. For a function $f : (\{0\} \cup \Sigma)^n \to \Gamma$ and an input $x \in (\{0\} \cup \Sigma)^n$ we define the certificate complexity $\mathrm{C}((f, w), x)$ to be the minimal weight of a certificate $S_1 \times \ldots \times S_n$ for f according to w, such that $x \in S_1 \times \ldots \times S_n$.

▶ **Definition 13** (Realization of Weight Schemes)**.** The weight-scheme defined by an integer-valued weight function $w : \Sigma \to \mathbb{N}$ is realized by $g_w : (\{0\} \cup \Sigma)^m \to (\{0\} \cup \Sigma)$ if:

- **(i)** For $i \in \Sigma$, there exists a collection of unambiguous certificates of size-$(w(i))$ for "$g_w(\cdot) = i$",
- **(ii)** $g_w(0^m) = 0$, and
- **(iii)** In order to prove "$g_w(0^m) \in S$" it is required to expose at least $w(S)$ coordinates of 0^m.

▶ **Lemma 14** (Weight-Scheme Implementation, [20])**.** *Let $w : \Sigma \to \mathbb{N}$ be an integer-valued weight function. Then, there exists a weight scheme associating natural numbers to non-empty subsets of $\{0\} \cup \Sigma$ that can be realized by a function $g_w : (\{0\} \cup \Sigma)^m \to (\{0\} \cup \Sigma)$ where $m = \max_i\{w_i\}$.*

[1] Note that a certificate for "$f(x) = 1$" for a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is always simple.

Proof. We define $g_w(x) = i$ iff the symbol i appears in the first $w(i)$ coordinates and i is the first non-zero symbol to appear in the string. We set $g_w(x) = 0$ if there is no such $i \in \Sigma$. (ii) holds trivially. For (i) note that the decision tree that queries the first $w(i)$ coordinates induces an unambiguous collection of certificates for "$g_w(\cdot) = i$". For (iii) we may assume without loss of generality that $S \neq (\{0\} \cup \Sigma)$ as otherwise the claim is trivial. Since we are proving that "$g_w(0^m) \in S$" and indeed $g_w(0^m) = 0$ it is required that $0 \in S$. It remains to show that it is required to expose the first $\max_{i \in \Sigma \setminus S} w(i)$ coordinates of the input to g_w. Let i be the element in $\Sigma \setminus S$ with maximal weight. Indeed, if one coordinate in the first $w(i)$ coordinates was not exposed, then it is still possible that $g_w(\cdot) = i$, as all coordinates that were exposed are equal to 0 and there is an unexposed position in the first $w(i)$ coordinates that might be marked with i. ◀

4.2 Composing Functions over Large Alphabet with Fractional Weights

Most of the results below are generalizations of arguments from [20]. However, since unlike [20] we deal with fractional weights, in addition to the total weight, we also need to take into account the number of coordinates in the intermediate certificates.

Let $f : (\{0\} \cup \Sigma)^N \to \{0, 1\}$, and let $w : \Sigma \to \mathbb{R}^+$. We treat the pair (f, w) as a "weighted function". Let $\mathcal{C}$ be an unambiguous collection of simple 1-certificates of size-s and weight at most W for (f, w). Let Σ_0 be a finite set that does not contain 0 and $w_0 : \Sigma_0 \to \mathbb{R}^+$. We define $(\tilde{f}, \tilde{w})$, where $\tilde{f} : (\{0\} \cup \Sigma \times \Sigma_0)^N \to (\{0\} \cup \Sigma_0)$ as follows. Denote by $\pi_1(x)$ and $\pi_2(x)$ the projection of $x \in (\{0\} \cup \Sigma \times \Sigma_0)^N$ to its $(\{0\} \cup \Sigma)^N$ coordinates and its $(\{0\} \cup \Sigma_0)^N$ coordinates respectively. The value of $\tilde{f}(x)$ is defined as follows.

> If $f(\pi_1(x)) = 0$, then set $\tilde{f}(x) := 0$. Otherwise, let $\mathcal{T} \in \mathcal{C}$ be the unique certificate for "$f(\cdot) = 1$" on $\pi_1(x)$. Read the corresponding coordinates of $\mathcal{T}$ from $\pi_2(x)$ and if all of them are equal to some $i \in \Sigma_0$, then set $\tilde{f}(x) := i$; otherwise set $\tilde{f}(x) := 0$.

Let $\tilde{w} : \Sigma \times \Sigma_0 \to \mathbb{R}^+$ be defined as $\tilde{w}(\sigma, i) = w(\sigma) \cdot w_0(i)$. The following lemma shows useful bounds on the certificates of the new function $\tilde{f}$ according to $\tilde{w}$.

▶ **Lemma 15** (From Boolean to Larger Output Alphabet). *Let $\tilde{f}$, f, $\tilde{w}$, w and w_0 be defined as above. Then,*

(B1) *There is an unambiguous collection of simple size-s certificates for "$\tilde{f}(\cdot) = i$" with weight at most $w_0(i) \cdot W$ according to $\tilde{w}$.*

(B2) *The certificate complexity of "$\tilde{f}(0^N) \neq i$" with respect to $\tilde{w}$ is at least $w_0(i) \cdot \mathrm{C}((f, w), 0^N)$.*

Proof.

(B1) The unambiguous collection of simple 1-certificates for f corresponds to unambiguous collection of simple i-certificates for $\tilde{f}$ by checking that each queried symbol has its Σ_0-part equals i. The weight of each certificate in the collection is at most $w_0(i) \cdot W$ as each coordinate weighs $w_0(i)$ times its "original" weight.

(B2) Fix $i \in \Sigma_0$. Assume we have a certificate for "$\tilde{f}(0^n) \neq i$". This is a cartesian product $S_1 \times \ldots \times S_N$ such that each S_i contains the 0 symbol and under which $\forall x \in S_1 \times \ldots \times S_N$ it holds that $\tilde{f}(x) \neq i$. Take $\hat{f}$ to be $\tilde{f}$ restricted only to input alphabet $\{0\} \cup (\Sigma \times \{i\})$. Then $S'_1 \times \ldots \times S'_n$ where $S'_j = S_j \cap (\{0\} \cup (\Sigma \times \{i\}))$ is a certificate for "$\hat{f}(0^N) \neq i$". Using property 3 in Definition 12, we show that $w(S'_j) \leq w(S_j)$. We consider two cases. If $S'_j = \{0\} \cup (\Sigma \times \{i\})$, then $w(S'_j) = 0 \leq w(S_j)$. Otherwise,

$$w(S'_j) = \max_{\sigma \in (\Sigma \times \{i\}) \setminus S'_j} w(\sigma) \leq \max_{\sigma \in (\Sigma \times \Sigma_0) \setminus S_j} w(\sigma) = w(S_j).$$

However, proving that "$\hat{f}(0^N) = 0$" is equivalent to proving that "$f(0^N) = 0$", except for the reweighing. Since each coordinate weighs according to $\tilde{w}$ at least $w_0(i)$ times its weight according to w, the weight of the certificate $S_1 \times \ldots \times S_n$ is at least $w_0(i) \cdot C((f,w),0^N)$. ◀

▶ **Lemma 16** (Composition Lemma). *Let $h : (\{0\} \cup [k])^n \to \{0,1\}$ with $h(0^n) = 0$, and let $w_0 : [k] \to \mathbb{R}^+$ such that (h, w_0) has an unambiguous collection of simple 1-certificates of size-k and (fractional) weight u, however any certificate for "$h(0^n) = 0$" is of (fractional) weight v.*

Let $\tilde{f}$ and $\tilde{w}$ be as defined above. Let $f' : (\{0\} \cup \Sigma \times [k])^{n \times N} \to \{0,1\}$ be defined by $f' = h \circ \tilde{f}$, and $w' : (\{0\} \cup \Sigma \times [k]) \to \mathbb{N}$ be equal to $\tilde{w}$. Then,

(A1) *(f', w') has an unambiguous collection of simple certificates 1-certificates with size at most sk and weight at most $u \cdot W$.*

(A2) $\mathrm{C}((f', w'), 0^{Nn}) \geq v \cdot \mathrm{C}((f,w), 0^N)$.

Proof.

(A1) Take the unambiguous collection $\mathcal{C}$ of simple 1-certificates for h of size-k and (fractional) weight u. For any certificate $\mathcal{T}$ from $\mathcal{C}$ replace the verification that some coordinate equals i with the simple certificate that the relevant N-length input of $\tilde{f}$ belongs to $\tilde{f}^{-1}(i)$. The cost of each such certificate to $\tilde{f}$ will be at most $W \cdot w_0(i)$ according to $\tilde{w} \equiv w'$. Thus, the overall cost will be $W \cdot u$, and the certificates will be of size at most sk. It is easy to verify that these certificates are unambiguous, since unambiguous collections of simple certificates are closed under composition.

(A2) Let $\mathcal{T}$ be a certificate for "$f'(0^{N \cdot n}) = 0$" of minimal weight (according to w'), and let $w_{\mathcal{T}}$ be its weight. Let $\mathcal{T}_1, \ldots, \mathcal{T}_n$ be the substrings of $\mathcal{T}$ of length N according to the composition of $h \circ \tilde{f}$. By Lemma 15[B1], if $\mathcal{T}_i$ certifies that "$\tilde{f}(0^N) \neq j$", then it costs at least $w_0(j) \cdot \mathrm{C}((f,w),0^N)$. We construct a certificate $\mathcal{H}$ for h from $\mathcal{T}$. If $\mathcal{T}_i$ certifies that $\tilde{f}(0^N) \neq j$ then $(\mathcal{H})_i \neq j$. More formally, let $\mathcal{H} = S_1 \times \ldots \times S_n$, where for $i \in [n]$ the set S_i consists of $\{0\}$ union with all j such that $\mathcal{T}_i$ *does not* certify that $\tilde{f}(0^N) \neq j$. Suppose by contradiction that $\mathcal{H}$ does not certify that "$h(0^n) = 0$". Then, there exists an input $y \in S_1 \times \ldots \times S_n$ (i.e., an input consistent with $\mathcal{H}$) such that $h(y) = 1$. Thus, there exist inputs $x^{(1)}, \ldots, x^{(n)}$ each of length N such that $\tilde{f}(x^{(i)}) = y_i$ and $\mathcal{T}_i$ is consistent with $x^{(i)}$, which shows that $\mathcal{T}$ is not a certificate for $h \circ \tilde{f}_{\tilde{w}}$. Thus, $\mathcal{H}$ is a certificate for $h(0^n) = 0$, and we get that

$$w_{\mathcal{T}} \geq w_0(\mathcal{H}) \cdot \mathrm{C}((f,w),0^N) = v \cdot \mathrm{C}((f,w),0^N).$$ ◀

Next, we show how to take any "gadget" h – a function over a constant number of symbols – with some gap between the $\mathrm{UC}_1(h)$ and $\mathrm{C}(h, 0^n)$, and convert it into an infinite family of functions with a polynomial separation between UC_1 and C.

▶ **Theorem 17** (From Gadgets to Boolean Unweighted Separations). *Let $u, v \in \mathbb{R}$, $k \in \mathbb{N}$ be constants such that $1 \leq k \leq u < v$. Let $h : (\{0\} \cup [k])^n \to \{0,1\}$ with $h(0^n) = 0$, and let $w_0 : [k] \to \mathbb{R}^+$ such that (h, w_0) has an unambiguous collection of simple 1-certificates of size-k and (fractional) weight u, however any certificate for "$h(0^n) = 0$" is of (fractional) weight v.*

Then, there exists an infinite family of Boolean functions $\{h'_m\}_{m \in \mathbb{N}}$ with

1. $\mathrm{UC}_1(h'_m) \leq \mathrm{poly}(m) \cdot u^m$
2. $\mathrm{C}(h'_m) \geq v^m$
3. *h'_m is defined over $\mathrm{poly}(m) \cdot \exp(O(m))$ many bits.*

Proof. We start by defining a sequence of weighted functions $\{(h_m, w_m)\}_{m\in\mathbb{N}}$ over large alphabet size with a polynomial gap between UC_1 and C. We then convert these functions into unweighted Boolean functions with the desired properties.

We take $h_1 := h$ and $w_1 := w_0$. For $m \geq 2$ we take (h_m, w_m) to be the composition of (h, w_0) with $(\tilde{h}_{m-1}, \tilde{w}_{m-1})$. Let $\Sigma_m = [k]^m$. Then, $h_m : (\{0\} \cup \Sigma_m)^{n^m} \to \{0,1\}$ and $w_m : (\{0\} \cup \Sigma_m) \to \mathbb{R}^+$. Using Lemma 16, we have that

(i) The maximal weight in w_m is at most $(w_{0,max})^m$, where $w_{0,max} := \max_i\{w_0(i)\}$.
(ii) There exists an unambiguous collection of simple 1-certificates of size k^m and weight at most u^m for (h_m, w_m).
(iii) $\mathrm{C}((h_m, w_m), \vec{0}) \geq v^m$.

Making Weights Integral

First, we modify the weights so that they will be integral. We take $w'_m(\cdot)$ to be $\lceil w_m(\cdot)\rceil$. Taking ceiling on the weights may only increase the certificate complexities. Thus, $\mathrm{C}((h_m, w'_m), \vec{0}) \geq v^m$. On the other hand, the weight of any certificate may only increase additively by its size, hence $\mathrm{UC}_1((h_m, w'_m)) \leq u^m + k^m \leq 2u^m$.

Eliminating Weights

Next, we convert the weighted function (h_m, w'_m) to an unweighted Boolean function h'_m with similar UC_1 and C complexities. First, we remove the weights by applying Lemma 14 (using the fact that w'_m is integer-valued). We define $h''_m = h_m \circ g_{w'_m}$. Lemma 14 implies that

$$\mathrm{C}(h''_m) \geq \mathrm{C}((h_m, w'_m)) \geq v^m$$

and

$$\mathrm{UC}_1(h''_m) \leq \mathrm{UC}_1((h_m, w'_m)) \leq 2 \cdot u^m.$$

Booleanizing

To make the inputs of the function h''_m Boolean we repeat the argument of Göös [20]. If f is a function $f : \Sigma^N \to \{0,1\}$, we may always convert it to a boolean function by composing it with some surjection $g_\Sigma : \{0,1\}^{\lceil \log|\Sigma|\rceil} \to \Sigma$. The following naive bounds will suffice for our purposes:

$$\mathcal{C}(f) \leq \mathcal{C}(f \circ g_\Sigma) \leq \mathcal{C}(f) \cdot \lceil \log|\Sigma|\rceil \quad \text{forall } \mathcal{C} \in \{\mathrm{UC}_1, \mathrm{C}\}. \tag{1}$$

In our final alphabet $\Sigma = \{0\} \cup [k]^m$, thus $h'_m = h''_m \circ g_\Sigma$ is a Boolean function with

$$\mathrm{C}(h'_m) \geq \mathrm{C}(h''_m) \geq v^m$$

and

$$\mathrm{UC}_1(h'_m) \leq \mathrm{UC}_1(h''_m) \cdot \lceil \log|\Sigma|\rceil \leq 2 \cdot u^m \cdot O(m \log k).$$

Input Length

The input length of h_m is n^m. By lemma 14, the input length of h''_m is at most $n^m \cdot (w^m_{0,max}+1)$. Thus the input length to h'_m is at most

$$O(\log(|\Sigma|) \cdot (n \cdot w_{0,max})^m) = O(m \cdot \log(k) \cdot (n \cdot w_{0,max})^m)$$ ◀

4.3 Gadgets Based on Projective Planes

We will use a reweighed version of the function constructed by Göös [20] based on projective planes as our gadget. Let us first recall the definition of a projective plane.

▶ **Definition 18** (Projective plane). A projective plane is a k-uniform hyper-graph with $n = k^2 - k + 1$ edges and n nodes with the following properties.

- Each node is incident on exactly k edges.
- For every two nodes, there exists a unique edge containing both.
- Every two edges intersect on exactly one node.

Given a projective plane, it follows from Hall's theorem that it is possible to assign an ordering to the edges incident to each vertex in a way that for each edge, its assigned order for each of its nodes is different. Namely, for each i, there are no two nodes for which their i-th incident edge is the same.

It is well-known that projective planes exist for every k such that $k-1$ is a prime power. Göös [20] introduced the following function $f : (\{0\} \cup \Sigma)^n \to \{0,1\}$ based on a projective plane, with $\Sigma = [k]$. We think of the inputs of f as a sequence of pointers, one for each node, where 0 is the Null pointer, and $i \in [k]$ is a pointer to the i-th edge on which the node is incident on. We set $f(x) = 1$ if there is an edge of the projective plane such that all its nodes point to it, and $f(x) = 0$ otherwise.

We will be interested in showing a gap between the certificate complexity of "$f(0^n) = 0$" and $\mathrm{UC}_1(f)$. However, the function as is, allows a certificate of size k for "$f(0^n) = 0$" matching its $\mathrm{UC}_1(f)$. One certificate for "$f(0^n) = 0$" is to pick an arbitrary edge of the projective plane, and certify that all its nodes have the Null pointer. This certifies "$f(0^n) = 0$" as every two edges in a projective plane intersect on a node. An unambiguous collection of size k certificates consists of picking for each edge all its nodes and ensuring that they point to that edge. This collection is unambiguous using the same property that every two edges intersect on one node.

In order to obtain a gadget with a gap between UC_1 and C, Göös introduced weights on the input alphabet of f. Each element $i \in \Sigma$ is assigned a weight $w(i)$, where the weights are intended to carry the following meaning: For every $i \in \Sigma$ it costs $w(i)$ for a certificate to assure that "$x_j = i$", and moreover 0 has the special property that it costs $\max_{i \in \Sigma} w(i)$ to assure that "$x_j = 0$" (as in Definition 12). In [20] each $i \in [k]$ is assigned a weight $w(i) = i$. Göös [20] implemented this weighting scheme specifically for the case when $w(i) := i$ via a weighting gadget $g_w : (\{0\} \cup \Sigma)^k \to (\{0\} \cup \Sigma)$ (as done in Lemma 14) and considering $f \circ g_w$. Our improvement comes from considering a different weighting scheme with fractional weights.

▶ **Claim 19** (Reweighing the Projective Plane). *Let f be defined as above, and let $w(i) := \frac{i}{(k+1)/2}$. Then, (f, w) has an unambiguous collection of simple 1-certificates of size k and weight k. Moreover, any certificate for $f(0^n) = 0$ is of weight at least $\frac{k^2-k+1}{(k+1)/2}$*

Proof. Göös [20, Claims 6 and 7] showed that with respect to the weight-function $w'(i) = i$, the function f has an unambiguous collection of simple 1-certificates of size-k and weight $(k \cdot (k+1))/2$. However, any certificate for "$f(0^n) = 0$" is of weight at least $k^2 - k + 1$.

From this, it is immediate that with respect to $w \equiv \frac{w'}{(k+1)/2}$, f has an unambiguous collection of simple 1-certificates of size-k and (fractional) weight $\frac{(k\cdot(k+1))/2}{(k+1)/2} = k$. However, any certificate for 0^n is of weight at least $\frac{k^2-k+1}{(k+1)/2}$. ◀

4.4 Putting Things Together

Given a gadget (h, w_0) such that h has unambiguous collection of simple 1-certificates of size-k and (fractional) weight u, however any certificate for 0^n is of (fractional) weight v, with $v > u > 1$ and $u \geq k$, Theorem 17 gives a polynomial separation between C and UC_1:

$$\mathrm{C}(h'_m) \geq v^m = (u^m)^{\log(v)/\log(u)} \geq \widetilde{\Omega}\left(\mathrm{UC}_1(h'_m)^{\log(v)/\log(u)}\right) . \tag{2}$$

We take h to be the projective plane function f described in Section 4.3 with $k = 8$, $n = k^2 - k + 1 = 57$ and weight function $w_0(i) = \frac{i}{(k+1)/2}$. By Claim 19, we have that with respect to w_0, h has an unambiguous collection of simple 1-certificates of size-k and weight $k = 8$. However, any certificate for 0^n is of weight $\frac{k^2-k+1}{(k+1)/2} = 38/3$. Plugging these values in Equation (2) we get a better separation:

$$\mathrm{C}(h'_m) \geq \widetilde{\Omega}\left(\mathrm{UC}_1(h'_m)^{\frac{\log(38/3)}{\log(8)}}\right) \geq \Omega(\mathrm{UC}_1(h'_m)^{1.22}) , \tag{3}$$

where the input length is $N \leq \mathrm{poly}(m) \cdot \exp(O(m))$. The lifting theorem of [22, 20] incurs a loss factor of $\log(N) = O(m)$ in the separation, however this is negligible compared to the $\mathrm{poly}(m) \cdot u^m$ versus v^m separation.

4.5 Further Improvements

Since our theorem is general in transforming a fractional weighted gadget into a polynomial separation, it is enough to only improve the gadget construction in order to improve the UC_1 vs C exponent. Indeed, even using the same gadget (the projective plane function of Göös) we can consider different weight function. Using computer search it seems that such reweighing is indeed better than our choice of w_0. However, the improvement is mild and currently we do not have a humanly verifiable proof for the lower bound on the certificate complexity of 0^n under the reweighing. Indeed, Göös relied on the fact that the weights were $w'(i) = i$ in order to present a simple proof of his lower bound on the certificate complexity of "$h(0^n) = 0$" according to w'. It seems though (we have verified this using computer-search for small values of k) that the best weights are attained by taking $w'(i) = i + 1$ and then reweighing by multiplying all weights by the constant $\alpha = \frac{1}{(k+3)/2}$, so that the unambiguous certificates for h will be of weights k. We leave proving a lower bound under this weight function as an open problem.

5 Attempting a Super-Quadratic Separation vs. Block Sensitivity

In this section, we describe why attempting to use Theorem 1 to get a super-quadratic separation between $\mathrm{bs}(f)$ and $\mathrm{s}(f)$ fails. In the process, we show some new lower bounds for $\mathrm{UC}_{\min}(f)$ and even for the one-sided non-negative degree measures.

One approach for the desired super-quadratic separation is to find a family of functions for which $\mathrm{bs}(f) \gg \mathrm{UC}_{\min}(f)$. In fact, by [28], it suffices to provide a family of functions for which $\mathrm{RC}(f) \gg \mathrm{UC}_{\min}(f)$ (as explained in Section 5.1). In Section 5.2, we show that even separating $\mathrm{RC}(f)$ from $\mathrm{UC}_{\min}(f)$ is impossible: we have $\mathrm{RC}(f) \leq 2\,\mathrm{UC}_{\min}(f) - 1$. This means our techniques do not give anything new for this problem. This is perhaps surprising, since $\mathrm{RC}(f)$ is similar to $\mathrm{C}(f)$, yet [20] showed a separation between $\mathrm{C}(f)$ and $\mathrm{UC}_{\min}(f)$.

5.1 A Separation Against $\mathrm{RC}(f)$ is Sufficient

[28] showed that a separation between $\mathrm{s}(f)$ and $\mathrm{RC}(f)$ implies an equal separation between $\mathrm{s}(f)$ and $\mathrm{bs}(f)$ (see Theorem 7). The key insight is that $\mathrm{bs}(f)$ becomes $\mathrm{RC}(f)$ when the function is composed enough times; this was observed by [40] and by [19]. This means that if we start with a function separating $\mathrm{s}(f)$ and $\mathrm{RC}(f)$ and compose it enough times, we should get a function with the same separation between $\mathrm{s}(f)$ and $\mathrm{RC}(f)$, but with the additional property that $\mathrm{bs}(f) \approx \mathrm{RC}(f)$.

5.2 But $\mathrm{RC}(f)$ Lower Bounds $\mathrm{UC}_{\min}(f)$

We would get a super-quadratic separation between $\mathrm{bs}(f)$ and $\mathrm{s}(f)$ if we had a super-linear separation between $\mathrm{RC}(f)$ and $\mathrm{UC}_{\min}(f)$. Unfortunately, this is impossible using our paradigm, as we now show. Actually, we can prove an even stronger statement, namely that $\mathrm{RC}(f) \leq (2\widetilde{\mathrm{avdeg}}^{+,\epsilon}_{\min}(f) - 1)/(1-4\epsilon)$. We note that this implies Theorem 8, because when $\epsilon = 0$, we have

$$\mathrm{RC}(f) \;\leq\; 2\,\mathrm{avdeg}^{+}_{\min}(f) - 1 \;\leq\; 2\,\mathrm{deg}^{+}_{\min}(f) - 1 \;\leq\; 2\,\mathrm{UC}_{\min}(f) - 1.$$

This stronger statement says that one-sided conical junta degree is lower bounded by two-sided randomized certificate complexity, which helps clarify the hierarchy of lower bounds for randomized algorithms.

The proof of the relationship $\mathrm{RC}(f) \leq (2\widetilde{\mathrm{avdeg}}^{+,\epsilon}_{\min}(f) - 1)/(1-4\epsilon)$ is somewhat technical; we leave it for Appendix A, and provide a cleaner proof (of $\mathrm{RC}(f) \leq 2\,\mathrm{UC}_{\min}(f) - 1$) below. One interesting thing to note about it is that it holds for partial functions as well, as long as the definition of $\widetilde{\mathrm{avdeg}}^{+,\epsilon}_{\min}(f)$ requires the approximating polynomial to evaluate to at most 1 on the entire Boolean hypercube.

Before providing the proof, we'll provide a warm up proof that $\mathrm{bs}(f) \leq 2\,\mathrm{UC}_{\min}(f)$.

▶ **Lemma 20.** *For all non-constant $f : \{0,1\}^n \to \{0,1\}$, we have $\mathrm{bs}(f) \leq 2\,\mathrm{UC}_{\min}(f) - 1$.*

Proof. Without loss of generality, we have $\mathrm{UC}_{\min}(f) = \mathrm{UC}_1(f)$. We also have $\mathrm{bs}_1(f) \leq \mathrm{C}_1(f) \leq \mathrm{UC}_1(f)$, so it remains to show that $\mathrm{bs}_0(f) \leq 2\,\mathrm{UC}_1(f) - 1$. Also without loss of generality, we assume that the block sensitivity of 0^n is $\mathrm{bs}(f)$ and that $f(0^n) = 0$.

Let $B_1, B_2, \ldots, B_{\mathrm{bs}(f)}$ be disjoint sensitive blocks of 0^n. Let U be an unambiguous collection of 1-certificates for f, each of size at most $\mathrm{UC}_1(f)$. For each $i \in [\mathrm{bs}(f)]$, we have $f(\vec{0}^{B_i}) = 1$, so there is some 1-certificate $p_i \in U$ such that p_i is consistent with $\vec{0}^{B_i}$. Since p_i is a 1-certificate, it is not consistent with $\vec{0}$, so it has a 1 bit (which must have index in B_i). Now, if $i \neq j$, the certificate p_i has a 1 inside B_i and only 0 or $*$ symbols outside B_i, and the certificate B_j has a 1 inside B_j and only 0 or $*$ symbols outside B_j; thus p_i and p_j are different. Since U is an unambiguous collection, p_i and p_j must conflict on some bit (with one of them assigning 0 and the other assigning 1), or else there would be an input consistent with both.

We construct a directed graph on vertex set $[\mathrm{bs}(f)]$ as follows. For each $i, j \in [\mathrm{bs}(f)]$ with $i \neq j$, we draw an arc from i to j if p_i has a 0 bit in a location where p_j has a 1 bit. It follows that for each pair $i, j \in [\mathrm{bs}(f)]$ with $i \neq j$, we either have an arc from i to j or else we have an arc from j to i (or both). The number of arcs in this graph is at least $\mathrm{bs}(f)(\mathrm{bs}(f)-1)/2$, so the average out degree is at least $(\mathrm{bs}(f)-1)/2$. Hence there is some vertex i with out degree at least $(\mathrm{bs}(f)-1)/2$. But this means p_i conflicts with $(\mathrm{bs}(f)-1)/2$ other certificates $p_{j_1}, p_{j_2}, \ldots, p_{j_{(\mathrm{bs}(f)-1)/2}}$ with p_i having a bit 0 and p_{j_k} having a 1-bit; however, two different

certificates p_{j_x} and p_{j_y} cannot both agree on a 1 bit, since the 1 bits of p_{j_x} must come from block B_{j_x} and the blocks are disjoint. This means p_i has at least $(\mathrm{bs}(f)-1)/2$ zero bits. It must also have at least one 1 bit. Thus $|p_i| \geq \mathrm{bs}(f)/2 + 1/2$, so $\mathrm{bs}(f) \leq 2\,\mathrm{UC}_{\min}(f) - 1$. ◀

We now generalize this lemma from bs to RC, proving Theorem 8. A further strengthening of the result can be found in Appendix A.

▶ **Theorem 8.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. Then* $\mathrm{RC}(f) \leq 2\,\mathrm{UC}_{\min}(f) - 1$.

Proof. Without loss of generality, we have $\mathrm{UC}_{\min}(f) = \mathrm{UC}_1(f)$. We also have $\mathrm{RC}_1(f) \leq \mathrm{C}_1(f) \leq \mathrm{UC}_1(f)$, so it remains to show that $\mathrm{RC}_0(f) \leq 2\,\mathrm{UC}_1(f) - 1$. Also without loss of generality, we assume that the randomized certificate of 0^n is $\mathrm{RC}(f)$ and that $f(0^n) = 0$.

We prove the theorem using the characterization of $\mathrm{RC}(f)$ as the fractional block sensitivity of f. Let $B_1, B_2, \ldots, B_m$ be minimal sensitive blocks of 0^n. Let $a_1, \ldots, a_m$ be weights assigned to blocks $B_1, \ldots, B_m$ such that

$$\sum_j a_j = \mathrm{RC}(f) \quad \text{, and} \quad \forall i \in [n] : \sum_{j : i \in B_j} a_j \leq 1 \,.$$

Let U be an unambiguous collection of 1-certificates for f, each of size at most $\mathrm{UC}_1(f)$. For each $i \in [m]$, we have $f(\vec{0}^{B_i}) = 1$, so there is some 1-certificate $p_i \in U$ such that p_i is consistent with $\vec{0}^{B_i}$. Since p_i is a 1-certificate, it is not consistent with $\vec{0}$, so it has a 1 bit (which must have index in B_i). Next, we show that if $i \neq j$, then p_i and p_j are different. Assume by contradiction that $p_i = p_j$, then p_i is a partial assignment that satisfy both $\vec{0}^{B_i}$ and $\vec{0}^{B_j}$, hence it must satisfy $\vec{0}^{B_i \cap B_j}$, but this means that $f(\vec{0}^{B_i \cap B_j}) = 1$ which contradicts the fact that both B_i and B_j are minimal sensitive blocks for $\vec{0}$.

We established that for any $i \neq j$, the partial assignments p_i and p_j are different. Since U is an unambiguous collection, p_i and p_j must conflict on some bit (with one of them assigning 0 and the other assigning 1), or else there would be an input consistent with both.

We construct a directed weighted graph on vertex set $[m]$ as follows. For each $i, j \in [m]$ with $i \neq j$, we draw an arc from i to j with weight $a_i \cdot a_j$, if p_i has a 0 bit in a location where p_j has a 1 bit. It follows that for each pair $i, j \in [m]$ with $i \neq j$, we either have an arc from i to j or else we have an arc from j to i (or both). The total weight of the arcs in this graph is

$$\begin{aligned}
\sum_{i<j} a_i \cdot a_j \cdot (|p_i^{-1}(1) \cap p_j^{-1}(0)| + |p_i^{-1}(0) \cap p_j^{-1}(1)|) &\geq \sum_{i<j} a_i \cdot a_j \\
&= \frac{1}{2} \cdot (\sum_i a_i)^2 - \frac{1}{2} \cdot \sum_i a_i^2 \\
&\geq \frac{1}{2} \cdot (\sum_i a_i)^2 - \frac{1}{2} \cdot \sum_i a_i \quad (a_i \leq 1) \\
&\geq \frac{1}{2} \cdot (\mathrm{RC}(f)^2 - \mathrm{RC}(f))
\end{aligned}$$

Note that by symmetry, the LHS equals

$$\sum_i a_i \cdot \sum_{j \neq i} a_j \cdot |p_i^{-1}(0) \cap p_j^{-1}(1)|.$$

Since $\sum_i a_i = \mathrm{RC}(f)$, by averaging,

$$\exists i : \ \frac{1}{2}(\mathrm{RC}(f) - 1) \leq \sum_{j \neq i} a_j \cdot |p_i^{-1}(0) \cap p_j^{-1}(1)|. \tag{4}$$

Next, we get a lower bound on $|p_i^{-1}(0)|$ from Eq. (4).

$$\begin{aligned}
\frac{1}{2}(\mathrm{RC}(f)-1) &\le \sum_{j\neq i} a_j \cdot |p_i^{-1}(0)\cap p_j^{-1}(1)| \\
&= \sum_{k:p_i(k)=0} \sum_{j:p_j(k)=1} a_j \\
&\le \sum_{k:p_i(k)=0} \sum_{j:k\in B_j} a_j && (p_j \text{ is consistent with } \vec{0}^{B_j}) \\
&\le |p_i^{-1}(0)|. && (\textstyle\sum_{j:k\in B_j} a_j \le 1 \text{ for all } k)
\end{aligned}$$

We showed that p_i has at least $(\mathrm{RC}(f)-1)/2$ zero bits. It must also have at least one 1 bit. Thus $|p_i| \ge \mathrm{RC}(f)/2 + 1/2$, so $\mathrm{RC}(f) \le 2\,\mathrm{UC}_{\min}(f) - 1$. ◀

We note that the relationships in Lemma 20 and Theorem 8 are tight.[2] Let k be any non-negative integer, we construct a function f on $n = 2k+1$ variables with $\mathrm{s}(f) = \mathrm{bs}(f) = \mathrm{RC}(f) = n$ and $\mathrm{UC}_{\min}(f) \le k+1$. This shows that the inequalities $\mathrm{bs}(f) \le 2\,\mathrm{UC}_{\min}(f) - 1$ and $\mathrm{RC}(f) \le 2\,\mathrm{UC}_{\min}(f) - 1$ are both tight for all values of $\mathrm{UC}_{\min}(f)$. We define the function f by describing a set of partial assignments $p_0, \ldots, p_{n-1}$ such that $f(x) = 1$ if and only if $\exists i : p_i \subseteq x$. Let $p = 0^k 1 *^k$. The assignments $p_0, \ldots, p_{n-1}$ are all possible cyclic-shifts of p, namely for $0 \le i \le k$, $p_i = 0^{k-i} 1 *^k 0^i$ and for $k+1 \le i \le 2k$ we have $p_i = *^{2k+1-i} 0^k 1 *^{i-1-k}$. It is easy to verify that any two different partial assignments p_i and p_j are not consistent with one another. Hence, $p_0, \ldots, p_{n-1}$ is an unambiguous collection of 1-certificates for f, each of size $k+1$, exhibiting that $\mathrm{UC}_{\min}(f) \le k+1$. On the other hand, $f(0) = 0$ and for all $i \in [n]$, we have $f(e_i) = 1$, showing that f has sensitivity n on the all-zeros input. Overall, we showed that $\mathrm{s}(f) = \mathrm{bs}(f) = \mathrm{RC}(f) = n = 2k+1$ while $\mathrm{UC}_{\min}(f) \le k$.

Acknowledgements. We would like to thank Mika Göös and Robin Kothari for many helpful discussions and for comments on a preliminary draft. We also thank the anonymous referees of ITCS for their comments.

References

1. Scott Aaronson. Quantum certificate complexity. *Journal of Computer and System Sciences*, 74(3):313–322, 2008. Computational Complexity 2003. doi:10.1016/j.jcss.2007.06.020.
2. Scott Aaronson, Shalev Ben-David, and Robin Kothari. Separations in query complexity using cheat sheets. *To appear in Proceedings of STOC 2016. arXiv preprint*, 2015. arXiv:1511.01937.
3. Andris Ambainis. Polynomial degree vs. quantum query complexity. *J. Comput. Syst. Sci.*, 72(2):220–238, 2006.
4. Andris Ambainis, Mohammad Bavarian, Yihan Gao, Jieming Mao, Xiaoming Sun, and Song Zuo. Tighter relations between sensitivity and other complexity measures. In *Automata, Languages, and Programming: 41st International Colloquium, ICALP 2014, Proceedings, Part I*, pages 101–113. Springer, 2014. doi:10.1007/978-3-662-43948-7_9.
5. Andris Ambainis, Martins Kokainis, and Robin Kothari. Nearly optimal separations between communication (or query) complexity and partitions. In *31st Conference on*

[2] We thank Mika Göös for helping to simplify this construction.

Computational Complexity (CCC 2016), volume 50 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:14, 2016. doi:10.4230/LIPIcs.CCC.2016.4.

6 Andris Ambainis and Krišjānis Prūsis. A tight lower bound on certificate complexity in terms of block sensitivity and sensitivity. In *Mathematical Foundations of Computer Science (MFCS 2014)*, pages 33–44. Springer, 2014. doi:10.1007/978-3-662-44465-8_4.

7 Andris Ambainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. Sensitivity versus certificate complexity of boolean functions. *arXiv preprint*, 2015. arXiv:1503.07691.

8 Andris Ambainis and Xiaoming Sun. New separation between s(f) and bs(f). *arXiv preprint*, 2011. arXiv:1108.3494.

9 Andris Ambainis and Jevgēnijs Vihrovs. Size of sets with small sensitivity: A generalization of simon's lemma. In *Theory and Applications of Models of Computation (TAMC 2015)*, pages 122–133. Springer, 2015. doi:10.1007/978-3-319-17142-5_12.

10 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald De Wolf. Quantum lower bounds by polynomials. *Journal of the ACM (JACM)*, 48(4):778–797, 2001. doi:10.1145/502090.502097.

11 Aleksandrs Belovs. Non-intersecting complexity. In *SOFSEM 2006: Theory and Practice of Computer Science*, pages 158–165. Springer, 2006. doi:10.1007/11611257_13.

12 Shalev Ben-David and Robin Kothari. Randomized query complexity of sabotaged and composed functions. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 60:1–60:14, 2016.

13 Meena Boppana. Lattice variant of the sensitivity conjecture. *arXiv preprint*, 2012. arXiv:1207.1824.

14 Yigal Brandman, Alon Orlitsky, and John Hennessy. A spectral lower bound technique for the size of decision trees and two-level and/or circuits. *IEEE Transactions on Computers*, 39(2):282–287, 1990. doi:10.1109/12.45216.

15 Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002. doi:10.1016/S0304-3975(01)00144-X.

16 Sourav Chakraborty, Raghav Kulkarni, Satyanarayana V Lokam, and Nitin Saurabh. Upper bounds on fourier entropy. *Theoretical Computer Science*, pages 771–782, 2016. Computing and Combinatorics 2015, TR13-052. doi:10.1016/j.tcs.2016.05.006.

17 Ehud Friedgut, Jeff Kahn, and Avi Wigderson. Computing graph properties by randomized subcube partitions. In *Randomization and approximation techniques in computer science (RANDOM 2002)*, pages 105–113. Springer, 2002. doi:10.1007/3-540-45726-7_9.

18 Justin Gilmer, Michal Koucký, and Michael E Saks. A new approach to the sensitivity conjecture. In *Conference on Innovations in Theoretical Computer Science (ITCS 2015)*, pages 247–254. ACM, 2015. doi:10.1145/2688073.2688096.

19 Justin Gilmer, Michael Saks, and Sudarshan Srinivasan. Composition limits and separating examples for some boolean function complexity measures. *Combinatorica*, pages 1–47, 2016. CCC 2013. doi:10.1007/s00493-014-3189-x.

20 Mika Göös. Lower bounds for clique vs. independent set. In *Foundations of Computer Science (FOCS 2015)*, pages 1066–1076. IEEE, 2015. TR15-012. doi:10.1109/FOCS.2015.69.

21 Mika Göös, T.S. Jayram, Toniann Pitassi, and Thomas Watson. Randomized communication vs. partition number. *Electronic Colloquium on Computational Complexity (ECCC)* TR15-169, 2015.

22 Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. In *Symposium on Theory of Computing (STOC), 2015*, pages 257–266, 2015.

23 Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. In *Foundations of Computer Science (FOCS 2015)*, pages 1077–1088. IEEE, 2015. TR15-050. doi:10.1109/FOCS.2015.70.

24 Parikshit Gopalan, Noam Nisan, Rocco A Servedio, Kunal Talwar, and Avi Wigderson. Smooth boolean functions are easy: Efficient algorithms for low-sensitivity functions. In *Conference on Innovations in Theoretical Computer Science (ITCS 2016)*, pages 59–70. ACM, 2016. doi:10.1145/2840728.2840738.

25 Parikshit Gopalan, Rocco Servedio, Avishay Tal, and Avi Wigderson. Degree and sensitivity: tails of two distributions. *arXiv preprint*, 2016. arXiv:1604.07432.

26 Pooya Hatami, Raghav Kulkarni, and Denis Pankratov. Variations on the sensitivity conjecture. *Theory of Computing, Graduate Surveys*, 4:1–27, 2011. doi:10.4086/toc.gs.2011.004.

27 Robin Kothari, David Racicot-Desloges, and Miklos Santha. Separating decision tree complexity from subcube partition complexity. In *Approximation, Randomization, and Combinatorial Optimization (RANDOM 2015)*, volume 40, pages 915–930. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPIcs.APPROX-RANDOM.2015.915.

28 Raghav Kulkarni and Avishay Tal. On fractional block sensitivity. *Chicago J. Theor. Comput. Sci.*, 2016, 2016.

29 Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek, and Mario Szegedy. Quantum query complexity of state conversion. In *Foundations of Computer Science (FOCS 2011)*, pages 344–353, 2011. doi:10.1109/FOCS.2011.75.

30 Gatis Midrijanis. Exact quantum query complexity for total boolean functions. *arXiv preprint*, 2004. arXiv:quant-ph/0403168.

31 Noam Nisan. Crew prams and decision trees. *SIAM Journal on Computing*, 20(6):999–1007, 1991. doi:10.1137/0220062.

32 Noam Nisan and Mario Szegedy. On the degree of Boolean functions as real polynomials. *Computational Complexity*, 4:301–313, 1994.

33 Ryan O'Donnell and Li-Yang Tan. A composition theorem for the fourier entropy-influence conjecture. In *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, pages 780–791, 2013.

34 Ryan O'Donnell, John Wright, Yu Zhao, Xiaorui Sun, and Li-Yang Tan. A composition theorem for parity kill number. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 144–154, 2014.

35 Ben W Reichardt. Reflections for quantum query algorithms. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms (SODA 2011)*, pages 560–569. SIAM, 2011. doi:10.1137/1.9781611973082.44.

36 Michael E. Saks and Avi Wigderson. Probabilistic boolean decision trees and the complexity of evaluating game trees. In *FOCS*, pages 29–38, 1986. doi:10.1109/SFCS.1986.44.

37 Petr Savicky. On determinism versus unambiquous nondeterminism for decision trees. *Electronic Colloquium on Computational Complexity (ECCC)* TR02-009, 2002.

38 Alexander A. Sherstov. Making polynomials robust to noise. *Theory of Computing*, 9:593–615, 2013.

39 Mario Szegedy. An $O(n^{0.4732})$ upper bound on the complexity of the gks communication game. *arXiv preprint*, 2015. arXiv:1506.06456.

40 Avishay Tal. Properties and applications of boolean function composition. In *Innovations in Theoretical Computer Science (ITCS 2013)*, pages 441–454, 2013. TR12-163. doi:10.1145/2422436.2422485.

41 Avishay Tal. On the sensitivity conjecture. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 38:1–38:13, 2016.

42 Ingo Wegener and Laszlo Zádori. A note on the relations between critical and sensitive complexity, 1988.
43 Mihalis Yannakakis. Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 43(3):441–466, 1991. STOC 1988. doi:10.1016/0022-0000(91)90024-Y.

A Lower Bound for Approximate Non-Negative Degree

Here we show that the lower bound in Theorem 8 holds even for one-sided average approximate non-negative degree, the smallest version of conical junta degree. This is saying that conical juntas, in all their forms, give a more powerful lower bound technique for randomized algorithms than $\mathrm{RC}(f)$.

▶ **Theorem 21.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a non-constant function, and let $\widetilde{\mathrm{avdeg}}^{+,\epsilon}_{\min}(f)$ denote the minimum average degree of a non-negative polynomial that approximates either f or its negation with error at most ϵ (see Section 2.6 for definitions). If $\epsilon < 1/4$, we have $\mathrm{RC}(f) \leq \frac{2\,\widetilde{\mathrm{avdeg}}^{+,\epsilon}_{\min}(f)-1}{1-4\epsilon}$.*

Proof. Let q be the non-negative approximating polynomial with average degree $\widetilde{\mathrm{avdeg}}^{+,\epsilon}_{\min}(f)$. Without loss of generality, we assume q approximates f rather than its negation. We can write $q \equiv \sum_{p\in\{0,1,*\}} w_p p$, so for any $x \in \{0,1\}^n$, we have

$$q(x) = \sum_{p\in\{0,1,*\}} w_p p(x) = \sum_{p:\, p\subseteq x} w_p,$$

where recall that w_p are non-negative weights given to partial assignments. This means for all $x \in \{0,1\}^n$, we know that

$$\left| f(x) - \sum_{p:\, p\subseteq x} w_p \right| \leq \epsilon, \qquad \sum_{p:\, p\subseteq x} w_p \leq 1, \qquad \text{and} \qquad \sum_{p:\, p\subseteq x} w_p|p| \leq \widetilde{\mathrm{avdeg}}^{+,\epsilon}_{\min}(f).$$

Now, consider the input $y \in \{0,1\}^n$ for which $\mathrm{RC}_y(f) = \mathrm{RC}(f)$. There are two cases: either y is a 0-input, or else y is a 1-input. If y is a 1-input, we use the fractional certificate complexity interpretation of $\mathrm{RC}_y(f)$: the value $\mathrm{RC}_y(f)$ is the minimum amount of weight that can be distributed to the bits of y such that every sensitive block of y contains bits of total weight at least 1. We assign to bit $i \in [n]$ the weight

$$\frac{1}{1-2\epsilon} \sum_{p:\, p\subseteq y, p_i\neq *} w_p.$$

Then each sensitive block $B \subseteq [n]$ for y satisfies $f(y^B) = 0$, so the sum of w_p over all $p \subseteq y$ that have support disjoint from B must be at most ϵ. Since the sum of w_p over all $p \subseteq y$ is at least $1-\epsilon$, there must be weight at least $1-2\epsilon$ assigned to partial assignments consistent with p whose support overlaps B. It follows that the total weight given to the bits in B is at least 1, which means this weighting is feasible. This means the total weight upper bounds $\mathrm{RC}_y(f)$, so

$$\mathrm{RC}(f) = \mathrm{RC}_y(f) \leq \frac{1}{1-2\epsilon} \sum_{i\in[n]} \sum_{p:\, p\subseteq y,\, p_i\neq *} w_p = \frac{1}{1-2\epsilon} \sum_{p:\, p\subseteq y} w_p|p| \leq \frac{\widetilde{\mathrm{avdeg}}^{+,\epsilon}_{\min}(f)}{1-2\epsilon}.$$

It remains to deal with the case where y is a 0-input. In this case, we use the fractional block sensitivity interpretation of $\mathrm{RC}_y(f)$: the value of $\mathrm{RC}_y(f)$ is the maximum amount of weight that can be distributed to the sensitive blocks of y such that every bit of y lies inside blocks of total weight at most 1. Without loss of generality, we can assume only minimal sensitive blocks are assigned weight (minimal sensitive blocks are sensitive blocks such that all their proper subsets are not minimal).

Let $\mathcal{B} := \{B \subseteq [n] : f(y^B) \neq f(y)\}$ be the set of sensitive blocks of y, and let $\mathcal{M} := \{B \in \mathcal{B} : \forall B' \subset B, B' \notin \mathcal{B}\}$ be the set of minimal sensitive blocks of y. Let $\{a_B\}_{B\in\mathcal{M}}$ with $a_B \in \mathbb{R}^+$ be the optimal weighting of the minimal sensitive blocks. This means $\sum_{B\in\mathcal{M}} a_B = \mathrm{RC}_y(f)$ and $\sum_{B\ni i} a_B \leq 1$ for all $i \in [n]$.

We have $\sum_{p\subseteq y} w_p \leq \epsilon$ and $\sum_{p\subseteq y^B} w_p \geq 1-\epsilon$ for all $B \in \mathcal{B}$. Thus, for any $B_1, B_2 \in \mathcal{M}$ with $B_1 \neq B_2$, we can write

$$2-2\epsilon \leq \sum_{p\subseteq y^{B_1}} w_p + \sum_{p\subseteq y^{B_2}} w_p = \sum_{p\subseteq y^{B_1} : p\not\subseteq y^{B_1\cup B_2}} w_p + \sum_{p\subseteq y^{B_2} : p\not\subseteq y^{B_1\cup B_2}} w_p + \sum_{p\in G} w_p + \sum_{p\in H} w_p,$$

where $G := \{p : p \subseteq y^{B_1}, p \subseteq y^{B_1\cup B_2}\}$ and $H := \{p : p \subseteq y^{B_2}, p \subseteq y^{B_1\cup B_2}\}$. The last two sums are equal to $\sum_{p\in G\cup H} w_p + \sum_{p\in G\cap H} w_p$. We have $\sum_{p\in G\cup H} w_p \leq \sum_{p\subseteq y^{B_1\cup B_2}} w_p \leq 1$. Also, any $p \in G\cap H$ satisfies $p \subseteq y^{B_1\cap B_2}$. Since $B_1 \neq B_2$ and they are both minimal sensitive blocks, we have $f(y^{B_1\cap B_2}) = 0$, so $\sum_{G\cap H} w_p \leq \sum_{p\subseteq y^{B_1\cap B_2}} w_p \leq \epsilon$. It follows that

$$\sum_{p\subseteq y^{B_1} : p\not\subseteq y^{B_1\cup B_2}} w_p + \sum_{p\subseteq y^{B_2} : p\not\subseteq y^{B_1\cup B_2}} w_p \geq 1-3\epsilon.$$

Note that the above sums are over disjoint sets, since if $p \subseteq y^{B_1}$ and $p \not\subseteq y^{B_1\cup B_2}$, then p must disagree with y^{B_2} on some bit inside B_2. If we split out the parts of the sums for which $p \subseteq y$, we get

$$\sum_{p\subseteq y} w_p + \sum_{p\subseteq y^{B_1} : p\not\subseteq y,\, p\not\subseteq y^{B_1\cup B_2}} w_p + \sum_{p\subseteq y^{B_2} : p\not\subseteq y,\, p\not\subseteq y^{B_1\cup B_2}} w_p \geq 1-3\epsilon.$$

Since $f(y) = 0$, the first sum is at most ϵ, so

$$\sum_{p\subseteq y^{B_1} : p\not\subseteq y,\, p\not\subseteq y^{B_1\cup B_2}} w_p + \sum_{p\subseteq y^{B_2} : p\not\subseteq y,\, p\not\subseteq y^{B_1\cup B_2}} w_p \geq 1-4\epsilon.$$

We now write the following.

$$\begin{aligned}
\mathrm{RC}(f)^2 - \mathrm{RC}(f) &= \sum_{B_1\in\mathcal{M}} a_{B_1} \sum_{B_2\in\mathcal{M}} a_{B_2} - \sum_{B_1\in\mathcal{M}} a_{B_1} \\
&\leq \sum_{B_1\in\mathcal{M}} a_{B_1} \sum_{B_2\in\mathcal{M}} a_{B_2} - \sum_{B_1\in\mathcal{M}} a_{B_1}^2 \\
&= \sum_{B_1\in\mathcal{M}} a_{B_1} \sum_{B_2\neq B_1} a_{B_2} \\
&\leq \frac{1}{1-4\epsilon} \sum_{B_1\in\mathcal{M}} a_{B_1} \sum_{B_2\neq B_1} a_{B_2} \\
&\quad \left(\sum_{p\subseteq y^{B_1} : p\not\subseteq y,\, p\not\subseteq y^{B_1\cup B_2}} w_p + \sum_{p\subseteq y^{B_2} : p\not\subseteq y,\, p\not\subseteq y^{B_1\cup B_2}} w_p \right) \\
&= \frac{2}{1-4\epsilon} \sum_{B_1\in\mathcal{M}} a_{B_1} \sum_{B_2\neq B_1} a_{B_2} \sum_{p\subseteq y^{B_1} : p\not\subseteq y,\, p\not\subseteq y^{B_1\cup B_2}} w_p,
\end{aligned}$$

where the second line follows because $a_{B_1} \leq 1$ for all $B_1 \in \mathcal{M}$.

Note that $\sum_{B_1 \in \mathcal{M}} a_{B_1} = \mathrm{RC}(f)$, so if we divide both sides by $\mathrm{RC}(f)$, the last line becomes a weighted average. It follows that there exists some minimal block B_1 such that

$$\begin{aligned}\mathrm{RC}(f) - 1 &\leq \frac{2}{1-4\epsilon} \sum_{B_2 \neq B_1} a_{B_2} \sum_{p \subseteq y^{B_1} : p \not\subseteq y,\, p \not\subseteq y^{B_1 \cup B_2}} w_p \\ &= \frac{2}{1-4\epsilon} \sum_{p \subseteq y^{B_1} : p \not\subseteq y} w_p \sum_{B_2 \neq B_1 : p \not\subseteq y^{B_1 \cup B_2}} a_{B_2}.\end{aligned}$$

Examine the inner summation above. Note that $y^{B_1 \cup B_2} = (y^{B_1})^{B_2 \setminus B_1}$. Since $p \subseteq y^{B_1}$, the condition $p \not\subseteq y^{B_1 \cup B_2}$ is equivalent to the support of p having non-empty intersection with $B_2 \setminus B_1$. Using $\mathrm{supp}(p)$ to denote the support of p, we have

$$\begin{aligned}\mathrm{RC}(f) - 1 &\leq \frac{2}{1-4\epsilon} \sum_{p \subseteq y^{B_1} : p \not\subseteq y} w_p \sum_{i \in \mathrm{supp}(p) \setminus B_1} \sum_{B_2 \in \mathcal{M} : i \in B_2} a_{B_2} \\ &\leq \frac{2}{1-4\epsilon} \sum_{p \subseteq y^{B_1} : p \not\subseteq y} w_p \sum_{i \in \mathrm{supp}(p) \setminus B_1} 1 \\ &= \frac{2}{1-4\epsilon} \sum_{p \subseteq y^{B_1} : p \not\subseteq y} w_p |\mathrm{supp}(p) \setminus B_1| \\ &\leq \frac{2}{1-4\epsilon} \sum_{p \subseteq y^{B_1} : p \not\subseteq y} w_p (|p| - 1) \\ &\leq \frac{2}{1-4\epsilon} \widetilde{\mathrm{avdeg}}^{+,\epsilon}_{\min}(f) - \frac{2}{1-4\epsilon} \sum_{p \subseteq y^{B_1} : p \not\subseteq y} w_p \\ &\leq \frac{2}{1-4\epsilon} \widetilde{\mathrm{avdeg}}^{+,\epsilon}_{\min}(f) - \frac{2}{1-4\epsilon} \left(\sum_{p \subseteq y^{B_1}} w_p - \sum_{p \subseteq y} w_p \right) \\ &\leq \frac{2}{1-4\epsilon} \widetilde{\mathrm{avdeg}}^{+,\epsilon}_{\min}(f) - \frac{2}{1-4\epsilon} (1 - \epsilon - \epsilon) \\ &\leq \frac{2}{1-4\epsilon} \widetilde{\mathrm{avdeg}}^{+,\epsilon}_{\min}(f) - \frac{2-4\epsilon}{1-4\epsilon},\end{aligned}$$

where the second line follows because the sum of a_B over all blocks $B \in \mathcal{M}$ containing a given element $i \in [n]$ is at most 1, and the fourth line follows because the conditions $p \subseteq y^{B_1}$ and $p \not\subseteq y$ imply that the support of p is not disjoint from B_1. Finally, we get

$$\mathrm{RC}(f) \leq \frac{2}{1-4\epsilon} \widetilde{\mathrm{avdeg}}^{+,\epsilon}_{\min}(f) - \frac{1}{1-4\epsilon} = \frac{2\,\widetilde{\mathrm{avdeg}}^{+,\epsilon}_{\min}(f) - 1}{1-4\epsilon},$$

as desired. ◀

Testing k-Monotonicity

Clément L. Canonne[1], Elena Grigorescu[2], Siyao Guo[3], Akash Kumar[4], and Karl Wimmer[5]

1 **Columbia University, New York, USA**
ccanonne@cs.columbia.edu
2 **Purdue University, West Lafayette, USA**
elena-g@purdue.edu
3 **New York University, New York, USA**
sg191@nyu.edu
4 **Purdue University, West Lafayette, USA**
akumar@purdue.edu
5 **Duquesne University, Pittsburgh USA**
wimmerk@duq.edu

Abstract

A Boolean k-*monotone* function defined over a finite poset domain $\mathcal{D}$ alternates between the values 0 and 1 at most k times on any ascending chain in $\mathcal{D}$. Therefore, k-monotone functions are natural generalizations of the classical *monotone* functions, which are the 1-*monotone* functions.

Motivated by the recent interest in k-monotone functions in the context of circuit complexity and learning theory, and by the central role that monotonicity testing plays in the context of property testing, we initiate a systematic study of k-monotone functions, in the property testing model. In this model, the goal is to distinguish functions that are k-monotone (or are close to being k-monotone) from functions that are far from being k-monotone.

Our results include the following:

1. We demonstrate a separation between testing k-monotonicity and testing monotonicity, on the hypercube domain $\{0,1\}^d$, for $k \geq 3$;
2. We demonstrate a separation between testing and learning on $\{0,1\}^d$, for $k = \omega(\log d)$: testing k-monotonicity can be performed with $2^{O(\sqrt{d}\cdot\log d\cdot\log 1/\varepsilon)}$ queries, while learning k-monotone functions requires $2^{\Omega(k\cdot\sqrt{d}\cdot 1/\varepsilon)}$ queries (Blais et al. (RANDOM 2015)).
3. We present a tolerant test for functions $f\colon [n]^d \to \{0,1\}$ with complexity independent of n, which makes progress on a problem left open by Berman et al. (STOC 2014).

Our techniques exploit the testing-by-learning paradigm, use novel applications of Fourier analysis on the grid $[n]^d$, and draw connections to distribution testing techniques.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Boolean Functions, Learning, Monotonicity, Property Testing

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.29

1 Introduction

A function $f\colon \mathcal{D} \to \{0,1\}$, defined over a finite partially ordered domain $(\mathcal{D}, \preceq)$ is said to be k-*monotone*, for some integer $k \geq 0$, if there does not exist $x_1 \preceq x_2 \preceq \ldots \preceq x_{k+1}$ in $\mathcal{D}$ such that $f(x_1) = 1$ and $f(x_i) \neq f(x_{i+1})$ for all $i \in [k]$. Note that 1-monotone functions are the classical *monotone* functions, satisfying $f(x_1) \leq f(x_2)$, whenever $x_1 \preceq x_2$.

Monotone functions have been well-studied on multiple fronts in computational complexity due to their natural structure. They have been celebrated for decades in the property


© Clément L. Canonne, Elena Grigorescu, Siyao Guo, Akash Kumar, and Karl Wimmer;
licensed under Creative Commons License CC-BY

testing literature [25, 20, 24, 10, 15, 14, 16], where we have recently witnessed ultimate results [33, 18, 6], in the circuit complexity literature, where we now have strong lower bounds [43, 44], and in computational learning, where we now have learning algorithms in numerous learning models [11, 3, 32, 46, 40, 41].

The generalized notion of k-monotonicity has also been studied in the context of circuit lower bounds for more than 50 years. In particular, Markov [36] showed that any k-monotone function (even with multiple outputs) can be computed using circuits containing only $\log k$ negation gates. The presence of negation gates appears to be a challenge in proving circuit lower bounds: "the effect of such gates on circuit size remains to a large extent a mystery" [29]. The recent results of Blais *et al.* [9] on circuit lower bounds have prompted renewed interest in understanding k-monotone functions from multiple angles, including cryptography, circuit complexity, learning theory, and Fourier analysis ([45, 27, 26, 35]).

Motivated by the exponential lower bounds on PAC learning k-monotone functions due to [9], we initiate the study of k-monotonicity in the closely related *Property Testing* model. In this model, given query access to a function, one must decide if the function is k-monotone, or is far from being k-monotone, by querying the input only in a small number of places.

1.1 Our results

We focus on testing k-monotonicity of Boolean functions defined over the d-dimensional hypegrid $[n]^d$, and the hypercube $\{0,1\}^d$. We begin our presentation with the results for the hypercube, in order to build intuition into the difficulty of the problem, while comparing our results with the current literature on testing monotonicity. Our stronger results concern the hypegrid $[n]^d$.

1.1.1 Testing k-monotonicity on the hypercube $\{0,1\}^d$

In light of the recent results of [33] that provide a $O(\sqrt{d})$-query tester for monotonicity, we first show that testing k-monotonicity is strictly harder than testing monotonicity on $\{0,1\}^d$, for $k \geq 3$.

▶ **Theorem 1.** *For $1 \leq k \leq d^{1/4}/2$, any one-sided non-adaptive tester for k-monotonicity of functions $f\colon \{0,1\}^d \to \{0,1\}$ must make $\Omega\big(d/k^2\big)^{k/4}$ queries.*

Both Theorem 1 and its proof generalize the $\Omega(d^{1/2})$ lower bound for testing monotonicity, due to Fischer *et al.* [24].

On the upper bounds side, while the monotonicity testing problem is providing numerous potential techniques for approaching this new problem [25, 20, 14, 10, 19, 33], most common techniques appear to resist generalizations to k-monotonicity. However, our upper bounds demonstrate a separation between testing and PAC learning k-monotonicity, for large enough values of $k = \omega(\log d)$.

▶ **Theorem 2.** *There exists a one-sided non-adaptive tester for k-monotonicity of functions $f\colon \{0,1\}^d \to \{0,1\}$ with query complexity $q(d,\varepsilon,k) = 2^{O\left(\sqrt{d}\cdot\log d\cdot\log\frac{1}{\varepsilon}\right)}$.*

Indeed, in the related PAC learning model, [9] shows that learning k-monotone functions on the hypercube requires $2^{\Omega(k\cdot\sqrt{d}\cdot 1/\varepsilon)}$ many queries.

We further observe that the recent non-adaptive and adaptive 2-sided lower bounds of [18, 6], imply the same bounds for k-monotonicity, using black box reductions. We summarize the state of the art for testing k-monotonicity on the hypercube in Table 1.

Table 1 Testing k-monotonicity of a function $f\colon \{0,1\}^d \to \{0,1\}$

	upper bound	1.s.-n.a. lower bound	2.s.-n.a. lower bound	2.s.-a. lower bound
$k=1$	$O\left(\sqrt{d}\right)$ [33]	$\Omega\left(d^{1/2}\right)$ [24]	$\Omega\left(d^{1/2-o(1)}\right)$ [18]	$\tilde{\Omega}\left(d^{1/4}\right)$ [6]
$k \geq 2$	$d^{O(k\sqrt{d})}$ [9], $d^{O(\sqrt{d})}$ Thm 2	$\Omega\left(d/k^2\right)^{k/4}$ Thm 1 $(k = O(d^{1/4}))$	$\Omega\left(d^{1/2-o(1)}\right)$	$\tilde{\Omega}\left(d^{1/4}\right)$

Table 2 Summary of our results: testing k-monotonicity of a function $f\colon [n]^d \to \{0,1\}$ (first two columns). The last column contains known bounds on monotonicity testing and is provided for comparison.

	General k	$k=2$	$k=1$ (monotonicity)
$d=1$	$\Theta\left(\frac{k}{\varepsilon}\right)$ 1.s.-n.a., $\tilde{O}\left(\frac{1}{\varepsilon^7}\right)$ 2.s.-n.a.	$O\left(\frac{1}{\varepsilon}\right)$ 1.s.-n.a.	$\Theta\left(\frac{1}{\varepsilon}\right)$ 1.s.-n.a.
$d=2$	$\tilde{O}\left(\frac{k^2}{\varepsilon^3}\right)$ 2.s.-n.a. (from below)	$\Theta\left(\frac{1}{\varepsilon}\right)$ 2.s.-a.	$\Theta\left(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}\right)$ 1.s.-n.a., $\Theta\left(\frac{1}{\varepsilon}\right)$ 1.s.-a.
$d \geq 3$	$\tilde{O}\left(\frac{1}{\varepsilon^2}\left(\frac{5kd}{\varepsilon}\right)^d\right)$ 2.s.-n.a., $2^{\tilde{O}(k\sqrt{d}/\varepsilon^2)}$ 2.s.-n.a.	$\tilde{O}\left(\frac{1}{\varepsilon^2}\left(\frac{10d}{\varepsilon}\right)^d\right)$ 2.s.-n.a., $2^{\tilde{O}(\sqrt{d}/\varepsilon^2)}$ 2.s.-n.a.	$O\left(\frac{d}{\varepsilon}\log\frac{d}{\varepsilon}\right)$ 1.s.-n.a.

1.1.2 Testing k-monotonicity on the hypergrid $[n]^d$

The remainder of the paper focuses on functions defined over the d-dimensional hypergrid domain $[n]^d$, where we denote by $(i_1, i_2, \ldots, i_d) \preceq (j_1, j_2, \ldots, j_d)$ the partial order in which $i_1 \leq j_1, i_2 \leq j_2, \ldots, i_d \leq j_d$. Testing monotonicity has received a lot of attention over the d-dimensional hypergrids [25, 21, 23, 5, 1, 28, 8, 15, 14, 16, 7], where the problem is well-understood, and we refer the reader the appendix of the full version for a detailed review on the state of the art in the area. We summarize our results on testing k-monotonicity over $[n]^d$ in Table 2.

1.1.2.1 Testing k-monotonicity on the line and the 2-dimensional grid

We begin with a study of functions $f\colon [n] \to \{0,1\}$. As before, note that 1-sided tests should always accept k-monotone functions, and so, they must accept unless they discover a violation to k-monotonicity in the form of a sequence $x_1 \preceq x_2 \preceq \ldots \preceq x_{k+1}$ in $[n]^d$, such that $f(x_1) = 1$ and $f(x_i) \neq f(x_{i+1})$. Therefore, lower bounds for 1-sided k-monotonicity testing must grow at least linearly with k. We show that this is indeed the case for both adaptive and non-adaptive tests, and moreover, we give a tight non-adaptive algorithm. Consequently, our results demonstrate that adaptivity does not help in testing k-monotonicity with one-sided error on the line domain.

▶ **Theorem 3.** *Any one-sided (possibly adaptive) tester for k-monotonicity of functions $f\colon [n] \to \{0,1\}$ must have query complexity $\Omega\left(\frac{k}{\varepsilon}\right)$.*

The upper bound generalizes the $O(1/\varepsilon)$ tester for monotonicity on the line.

▶ **Theorem 4.** *There exists a one-sided non-adaptive tester for k-monotonicity of functions $f\colon [n] \to \{0,1\}$ with query complexity $q(n, \varepsilon, k) = O\left(\frac{k}{\varepsilon}\right)$.*

Testing with 2-sided error, however, does not require a dependence on k. In fact the problem has been well-studied in the machine learning literature in the context of testing/learning "union of intervals" [31, 4], and in testing geometric properties, in the

context of testing surface area [34, 38],[1] resulting in an $O(1/\varepsilon^{7/2})$-query algorithm. Namely, the starting point of [4] (later improved by [34]) is a "Buffon Needle's"-type argument, where the crucial quantity to analyze is the noise sensitivity of the function, that is the probability that a randomly chosen pair of nearby points cross a "boundary" – i.e., have different values. (Moreover, the algorithm of [4] works in the *active testing* setting: it only requires a weaker access model that the standard query model).

We provide an alternate proof of a poly$(1/\varepsilon)$ bound (albeit with a worse exponent) that reveals a surprising connection with *distribution testing*, namely with the problem of estimating the support size of a distribution.

▶ **Theorem 5.** *There exists a two-sided non-adaptive tester for k-monotonicity of functions $f\colon [n] \to \{0,1\}$ with query complexity $q(n,\varepsilon,k) = \tilde{O}\big(1/\varepsilon^7\big)$,* independent of k.

An immediate implication of Theorem 5 is that one can test even $n^{1-\alpha}$-monotonicity of $f\colon [n] \to \{0,1\}$, for every $\alpha > 0$, with a constant number of queries. Hence, there is a separation between 1-sided and 2-sided testing, for $k = \omega(1)$.

Turning to the 2-dimensional grid, we show that 2-monotone functions can be tested with the minimum number of queries one could hope for:

▶ **Theorem 6.** *There exists a two-sided adaptive tester for 2-monotonicity of functions $f\colon [n]^2 \to \{0,1\}$ with query complexity $q(n,\varepsilon) = O\big(\frac{1}{\varepsilon}\big)$.*

We also discuss possible generalizations of Theorem 6 to general k or d in the full version.

1.1.2.2 Testing k-monotonicity on $[n]^d$, tolerant testing, and distance approximation

Moving to the general grid domain $[n]^d$, we show that k-monotonicity is testable with poly$(1/\varepsilon, k)$ queries in constant-dimension grids.

▶ **Theorem 7.** *There exists a non-adaptive tester for k-monotonicity of functions $f\colon [n]^d \to \{0,1\}$ with query complexity $q(n,d,\varepsilon,k) = \min(\tilde{O}\Big(\frac{1}{\varepsilon^2}\left(\frac{5kd}{\varepsilon}\right)^d\Big), 2^{\tilde{O}\left(k\sqrt{d}/\varepsilon^2\right)})$.*

In fact, we obtain more general testing algorithms than in Theorem 7, namely our results hold for *tolerant* testers (as we define next).

The notion of tolerant testing was first introduced in [42] to account for the possibility of noisy data. In this notion, a test should accept inputs that are ε_1-close to the property, and reject inputs that are ε_2-far from the property, where ε_1 and ε_2 are given parameters. Tolerant testing is intimately connected to the notion of distance approximation: given tolerant testers for every $(\varepsilon_1, \varepsilon_2)$, there exists an algorithm that estimates the distance to the property within any (additive) ε, while incurring only a $\tilde{O}\big(\log\frac{1}{\varepsilon}\big)$ factor blow up in the number of queries. Furthermore, [42] shows that both tolerant testing and distance approximation are no harder than agnostic learning. We prove the following general result.

▶ **Theorem 8.** *There exist*

- *a non-adaptive (fully) tolerant tester for k-monotonicity of functions $f\colon [n]^d \to \{0,1\}$ with query complexity $q(n,d,\varepsilon_1,\varepsilon_2,k) = \tilde{O}\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\left(\frac{5kd}{\varepsilon_2-\varepsilon_1}\right)^d\right)$;*
- *a non-adaptive tolerant tester for k-monotonicity of functions $f\colon [n]^d \to \{0,1\}$ with query complexity $q(n,d,\varepsilon_1,\varepsilon_2,k) = 2^{\tilde{O}\left(k\sqrt{d}/(\varepsilon_2-3\varepsilon_1)^2\right)}$, under the restriction that $\varepsilon_2 > 3\varepsilon_1$.*

[1] We thank Eric Blais for mentioning the connection, and pointing us to these works.

To the best of our knowledge, the only previous results for tolerant testing for monotonicity on $[n]^d$ are due to Fattal and Ron [22]. They give both additive and multiplicative distance approximations algorithms, and obtain $O(d)$-multiplicative and ε-additive approximations with query complexity poly$(\frac{1}{\varepsilon})$. While very efficient, there results only give fully tolerant testers for dimensions $d = 1$ and $d = 2$. Our results generalize the work of [22] showing existence of tolerant testers for k-monotonicity (and hence for monotonicity) for any dimension $d \geq 1$, and any $k \geq 1$, but paying the price in the query complexity.

As a consequence to Theorem 8, we make progress on an open problem of Berman *et al.* [7], as explained next.

1.1.2.3 Testing k-monotonicity under L_p distance

The property of being a monotone Boolean function has a natural extension to real-valued functions. Indeed, a real-valued function defined over a finite domain D is monotone if $f(x) \leq f(y)$ whenever $x \preceq y$. For real-valued functions the more natural notion of distance is L_p distance, rather than Hamming distance. The study of monotonicity has been extended to real-valued functions in a recent work by Berman *et al.* [7]. They give tolerant testers for grids of dimension $d = 1$ and $d = 2$, and leave open the problem of extending the results to general d, as asked explicitly at the recent Sublinear Algorithms Workshop 2016 [47].

We make progress towards solving this open problem, by combining our Theorem 8 with a reduction from L_p testing to Hamming testing inspired by [7]. This reduction relates L_1-distance of a function $f\colon [n]^d \to [0,1]$ to monotonicity to *Hamming* distance to monotonicity of a "rounded" function $\tilde{f}\colon [n]^d \times [m] \to \{0,1\}$, essentially trading the range for an extra dimension (where m is a rounding parameter to be suitably chosen). Moreover, simulating query access to $\tilde{f}$ can be performed efficiently given query access to f.

▶ **Theorem 9.** *There exists a non-adaptive tolerant L_1-tester for monotonicity of functions $f\colon [n]^d \to [0,1]$ with query complexity*

- $\tilde{O}\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\left(\frac{5d}{\varepsilon_2-\varepsilon_1}\right)^d\right)$*, for any* $0 \leq \varepsilon_1 < \varepsilon_2 \leq 1$*;*
- $2^{\tilde{O}\left(\sqrt{d}/(\varepsilon_2-3\varepsilon_1)^2\right)}$*, for any* $0 \leq 3\varepsilon_1 < \varepsilon_2 \leq 1$*.*

1.2 Proofs overview and technical contribution

Structural properties and the separation between testing and learning on $\{0,1\}^d$

We first observe that basic structural properties, such as *extendability* (i.e. the feature that a function that is monotone on a sub-poset of $[n]^d$ can be extended into a monotone function on the entire poset domain), and properties of the *violation graph* (i.e., the graph whose edges encode the violations to monotonicity), extend easily to k-monotonicity (see the full version of the paper). These properties help us to argue the separation between testing and learning (Theorem 2). However, unlike the case of monotonicity testing, these properties do not seem to be enough for showing upper bounds that grow polynomially in d.

Grid coarsening and testing by implicit/explicit learning

One pervading technique, which underlies all the hypergrid upper bounds in this work, is that of *gridding*: i.e., partitioning the domain into "blocks" whose size no longer depends on the main parameter of the problem, n. This technique generalizes the approach of [22] who performed a similar gridding for dimension $d = 2$. By simulating query access to the "coarsened" version of the unknown function (with regard to these blocks), we are able to

leverage methods such as testing-by-learning (either fully or partially learning the function), or reduce our testing problem to a (related) question on these nicer "coarsenings." (The main challenge here lies in providing efficient and consistent oracle access to the said coarsenings.)

At a high-level, the key aspect of k-monotonicity which makes this general approach possible is reminiscent of the concept of *heredity* in property testing. Specifically, we rely upon the fact that "gridding preserves k-monotonicity:" if f *is* k-monotone, then so will be its coarsening g – but now g is much simpler to handle. This allows us to trade the domain $[n]^d$ for what is effectively $[m]^d$, with $m \ll n$. We point out that this differs from the usual paradigm of *dimension reduction*: indeed, the latter would reduce the study of a property of functions on $[n]^d$ to that of functions on $[n]^{d'}$ for $d' \ll d$ (usually even $d' = 1$) by projecting f on a lower-dimensional domain. In contrast, we do not take the dimension down, but instead reduce the size of the *alphabet*.Moreover, it is worth noting that this gridding technique is also orthogonal to that of *range reduction*, as used e.g. in [20]. Indeed, the latter is a reduction of the range of the function from $[R]$ to $\{0, 1\}$, while gridding is only concerned about the domain size.

Estimating the support of distributions

Our proof of the poly$(1/\varepsilon)$ upper bound for testing k-monotonicity on the line (Theorem 5) rests upon an unexpected connection to *distribution testing*, namely to the question of support size estimation of a probability distribution. In more detail, we describe how to reduce k-monotonicity testing to the support size estimation problem in (a slight modification of) the *Dual access model* introduced by Canonne and Rubinfeld [13], where the tester is granted samples from an unknown distribution as well as query access to its probability mass function.

For our reduction to go through, we first describe how any function $f\colon [n] \to \{0, 1\}$ determines a probability distribution D_f (on $[n]$), whose effective support size is directly related to the k-monotonicity of f. We then show how to implement dual access to this D_f from queries to f: in order to avoid any dependence on k and n in this step, we resort both to the gridding approach outlined above (allowing us to remove n from the picture) and to a careful argument to "cap" the values of D_f returned by our simulated oracle. Indeed, obtaining the exact value of $D_f(x)$ for arbitrary x may require $\Omega(k)$ queries to f, which we cannot afford; instead, we argue that only returning $D_f(x)$ whenever this value is "small enough" is sufficient. Finally, we show that implementing this "capped" dual access oracle is possible with no dependence on k whatsoever, and we can now invoke the support size estimation algorithm of [13] to conclude.

Fourier analysis on the hypergrid

We give an algorithm for fully tolerantly testing k-monotonicity whose query complexity in exponential in d. We also describe an alternate tester (with a slightly worse tolerance guarantee) whose query complexity is instead exponential in $\widetilde{O}(k\sqrt{d})$ for constant distance parameters. As mentioned above, we use our gridding approach combined with tools from learning theory. Specifically, we employ an agnostic learning algorithm of [30] using polynomial regression. Our coarsening methods allow us to treat the domain as if it were $[m]^d$ for some m that is independent of n. To prove that this agnostic learning algorithm will succeed, we turn to Fourier analysis over $[m]^d$. We extend the bound on average sensitivity of k-monotone functions over the Boolean hypercube from [9] to the hypergrid, and we show that this result implies that the Fourier coefficients are concentrated on "simple" functions.

1.3 Discussion and open problems

This is the first work to study k-monotonicity, a natural and well-motivated generalization of monotonicity. Hence this work opens up many intriguing questions in the area of property testing, with potential applications to learning theory, circuit complexity and cryptography.

As previously mentioned, the main open problem prompted by our work is the following:

Can k-monotonicity on the hypercube $\{0,1\}^d$ be tested with $\mathrm{poly}(d^k)$ queries?

A natural 1-sided tester for k-monotonicity is a *chain tester*: it queries points along a random chain, and rejects only if it finds a violation to k-monotonicity, in the form of a sequence $x_1 \preceq x_2 \preceq \ldots \preceq x_{k+1}$ in $\{0,1\}^d$, such that $f(x_1) = 1$ and $f(x_i) \neq f(x_{i+1})$. In particular, the testers in [25, 14, 19, 33] all directly imply a chain tester. We conjecture that there exists a chain tester for k-monotonicity that succeeds with probability $d^{-O(k)}$.

Another important open question concerns the hypergrid domain, and in particular it pushes for a significant strengthening of Theorem 7 and Theorem 9:

Can k-monotonicity on the hypergrid $[n]^d$ be (tolerantly) tested with $2^{o_k(\sqrt{d})}$ queries?

Answering this question would imply further progress on the L_1-testing question for monotonicity, left open in [7, 47].

There also remains the question of establishing two-sided lower bounds that would go beyond those of monotonicity. Specifically:

Is there an $d^{\Omega(k)}$-query two-sided *lower bound for k-monotonicity on the hypercube $\{0,1\}^d$?*

In this work we also show surprising connections to distribution testing (e.g. in the proof of Theorem 5), and to testing union of intervals and testing surface area. An intriguing direction is to generalize this connection to union of intervals and surface area in higher dimensions, to leverage or gain insight on k-monotonicity on the d-dimensional hypergrid.

Finally, while we only stated here a few directions, we emphasize that every question that is relevant to monotonicity is also relevant and interesting in the case of k-monotonicity.

1.4 Related work

As mentioned, k-monotonicity has deep connections with the notion of *negation complexity* of functions, which is the minimum number of negation gates needed in a circuit to compute a given function. The power of negation gates is intriguing and far from being understood in the context of circuit lower bounds. Quoting from Jukna's book [29], *the main difficulty in proving nontrivial lower bounds on the size of circuits using AND, OR, and NOT is the presence of NOT gates: we already know how to prove even exponential lower bounds for monotone functions if no NOT gates are allowed. The effect of such gates on circuit size remains to a large extent a mystery.*

This gap has motivated the study of circuits with *few* negations. Two notable works successfully extend lower bounds in the monotone setting to negation-limited setting: in [2], Amano and Maruoka show superpolynomial circuit lower bounds for $(1/6)\log\log n$ negations using the Clique function; and recently the breakthrough work of Rossman [45] establishes circuit lower bounds for NC^1 with roughly $\frac{1}{2}\log n$ negations by drawing upon his lower bound for monotone NC^1.

The divide between the understanding of monotone and non-monotone computation exists in general: while we usually have a fairly good understanding of the monotone case, many things get murky or fail to hold even when a single negation gate is allowed. In order to get a better grasp on negation-limited circuits, a body of recent work has been considering this model in various contexts: Blais *et al.* [9] study negation-limited circuits from a computational learning viewpoint, Guo *et al.* [27] study the possibility of implementing cryptographic primitives using few negations, and Lin and Zhang [35] are interested in verifying whether some classic Boolean function conjectures hold for the subset of functions computed by negation-limited circuits.

Many of these results implicitly or explicitly rely on a simple but powerful tool: the decomposition of negation-limited circuits into a composition of some "nice" function with monotone components. Doing so enables one to apply results on separate monotone components, and finally to carefully combine the outcomes (e.g., [26]). Though these techniques can yield results for as many as $O(\log n)$ negations, they also leave open surprisingly basic questions:

- [9] Can we have an efficient weak learning algorithm for functions computed by circuits with a *single* negation?
- [27] Can we obtain pseudorandom generators when allowing only a *single* negation?

In contexts where the circuit size is not the quantity of interest, the equivalent notion of 2-monotone functions is more natural than that of circuits allowing only one negation. Albeit seemingly simple, even the class of 2-monotone functions remains largely a mystery: as exemplified above, many basic yet non-trivial questions, ranging from the structure of their Fourier spectrum to their expressive power of k-monotone functions, remain open.

1.5 Organization of the paper

After recalling some notations and definitions in section 2, we consider the case of functions on the line in section 3, focusing on the proof of the two-sided upper bound of Theorem 3.

In section 4 we present our general algorithms for k-monotonicity on the hypergrid $[n]^d$, for arbitrary k and d. We prove Theorem 8 in two parts. We establish its first item (general tolerant testing algorithm with exponential dependence in d) in subsection 4.1 (Proposition 22). The second item (with query complexity exponential in $k\sqrt{d}$) is proven in subsection 4.2, where we analyze the Fourier-based tolerant tester of Proposition 31.

Our results on the Boolean hypercube, the two-dimensional grid, as well as some structural results and applications to tolerant L_1-testing of monotonicity have been left out of this short version, and are deferred to the full version of the paper [12].

2 Preliminaries

We denote by log the binary logarithm, and use $\tilde{O}(\cdot)$ to hide polylogarithmic factors in the argument (so that $\tilde{O}(f) = O(f \log^c f)$ for some $c \geq 0$).

Given two functions $f, g\colon \mathcal{X} \to \mathcal{Y}$ on a finite domain $\mathcal{X}$, we write $\mathrm{dist}(f,g)$ for the (normalized) Hamming distance between them, i.e.

$$\mathrm{dist}(f,g) = \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} \mathbb{1}_{\{f(x)\neq g(x)\}} = \Pr_{x\sim\mathcal{X}}[\,f(x)\neq g(x)\,]$$

where $x \sim \mathcal{X}$ refers to x being drawn from the uniform distribution on $\mathcal{X}$. A *property* of functions from $\mathcal{X}$ to $\mathcal{Y}$ is a subset $\mathcal{P} \subseteq \mathcal{X}^{\mathcal{Y}}$ of these functions; we define the distance of a function f to $\mathcal{P}$ as the minimum distance of f to any $g \in \mathcal{P}$:

$$\mathrm{dist}(f,\mathcal{P}) = \inf_{g\in\mathcal{P}} \mathrm{dist}(f,g)\,.$$

For some of our applications, we will also use another notion of distance specific to real-valued functions, the L_1 distance (as introduced in the context of property testing in [7]). For $f, g\colon \mathcal{X} \to [0,1]$, we write

$$L_1(f,g) = \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} |f(x)-g(x)| = \mathbb{E}_{x\sim\mathcal{X}}[|f(x)-g(x)|] \in [0,1]$$

and extend the definition to $L_1(f,\mathcal{P})$, for $\mathcal{P} \subseteq \mathcal{X}^{[0,1]}$, as before.

Property testing

We recall the standard definition of testing algorithms, as well as some terminology:

▶ **Definition 10.** Let $\mathcal{P}$ be a property of functions from $\mathcal{X}$ to $\mathcal{Y}$. A *q-query testing algorithm for $\mathcal{P}$* is a randomized algorithm $\mathcal{T}$ which takes as input $\varepsilon \in (0,1]$ as well as query access to a function $f\colon \mathcal{X} \to \mathcal{Y}$. After making at most $q(\varepsilon)$ queries to the function, $\mathcal{T}$ either outputs ACCEPT or REJECT, such that the following holds:

- if $f \in \mathcal{P}$, then $\mathcal{T}$ outputs ACCEPT with probability at least 2/3; (Completeness)
- if $\mathrm{dist}(f,\mathcal{P}) \geq \varepsilon$, then $\mathcal{T}$ outputs REJECT with probability at least 2/3; (Soundness)

where the probability is taken over the algorithm's randomness. If the algorithm only errs in the second case but accepts any function $f \in \mathcal{P}$ with probability 1, it is said to be a *one-sided* tester; otherwise, it is said to be *two-sided*. Moreover, if the queries made to the function can only depend on the internal randomness of the algorithm, but not on the values obtained during previous queries, it is said to be *non-adaptive*; otherwise, it is *adaptive*.

Additionally, we will also be interested in *tolerant* testers – roughly, algorithms robust to a relaxation of the first item above:

▶ **Definition 11.** Let $\mathcal{P}$, $\mathcal{X}$, and $\mathcal{Y}$ be as above. A *q-query tolerant testing algorithm for $\mathcal{P}$* is a randomized algorithm $\mathcal{T}$ which takes as input $0 \leq \varepsilon_1 < \varepsilon_2 \leq 1$, as well as query access to a function $f\colon \mathcal{X} \to \mathcal{Y}$. After making at most $q(\varepsilon_1,\varepsilon_2)$ calls to the oracle, $\mathcal{T}$ outputs either ACCEPT or REJECT, such that the following holds:

- if $\mathrm{dist}(f,\mathcal{P}) \leq \varepsilon_1$, then $\mathcal{T}$ outputs ACCEPT with probability at least 2/3; (Completeness)
- if $\mathrm{dist}(f,\mathcal{P}) \geq \varepsilon_2$, then $\mathcal{T}$ outputs REJECT with probability at least 2/3; (Soundness)

where the probability is taken over the algorithm's randomness. The notions of one-sidedness and adaptivity of Theorem 10 extend to tolerant testing algorithms as well.

Note that as stated, in both cases the algorithm "knows" $\mathcal{X}, \mathcal{Y}$, and $\mathcal{P}$; so that the query complexity q can be parameterized by these quantities. More specifically, when considering $\mathcal{X} = [n]^d$ and the property $\mathcal{P}$ of k-monotonicity, we will allow q to depend on n, d, and k. Finally, we shall sometimes require a probability of success $1-\delta$ instead of the (arbitrary) constant 2/3; by standard techniques, this can be obtained at the cost of a multiplicative $O(\log(1/\delta))$ in the query complexity.

PAC and agnostic learning [48]

A learning algorithm $\mathcal{A}$ for a *concept class* $\mathcal{C}$ of functions $f\colon \mathcal{X} \to \mathcal{Y}$ (under the uniform distribution) is given parameters $\varepsilon, \delta > 0$ and sample access to some target function $f \in \mathcal{C}$ *via* labeled samples $\langle x, f(x)\rangle$, where x is drawn uniformly at random from $\mathcal{X}$. The algorithm should output a hypothesis $h\colon \mathcal{X} \to \mathcal{Y}$ such that $\operatorname{dist}(h, f) \leq \varepsilon$ with probability at least $1-\delta$. The algorithm is *efficient* if it runs in time $\operatorname{poly}(n, 1/\varepsilon, 1/\delta)$. If $\mathcal{A}$ must output $h \in \mathcal{C}$ we say it is a *proper learning algorithm*, otherwise, we say it is an *improper learning* one.

Moreover, if $\mathcal{A}$ still succeeds when f does not actually belong to $\mathcal{C}$, we say it is an *agnostic learning algorithm*. Specifically, the hypothesis function h that it outputs must satisfy $\operatorname{dist}(f, g) \leq \textsc{opt}_f + \varepsilon$ with probability at least $1-\delta$, where $\textsc{opt}_f = \min_{g\in\mathcal{C}} \operatorname{dist}(f, g)$.

3 On the line

In this section we focus on testing k-monotonicity on the line, that is of functions $f\colon [n] \to \{0,1\}$. Our results include Theorem 4, which establishes that this can be done non-adaptively with one-sided error, with only $O(k/\varepsilon)$ queries; complemented by Theorem 3, which shows that this is the best one can hope for if we insist on one-sidedness. Due to space constraints, we only prove here Theorem 5 (restated below), which shows that – perhaps unexpectedly – *two-sided* algorithms, even non-adaptive, can break this barrier and test k-mononicity with *no* dependence on k. The proofs of Theorem 4 and Theorem 3 can be found in the full version.

▶ **Theorem 5.** *There exists a two-sided non-adaptive tester for k-monotonicity of functions $f\colon [n] \to \{0,1\}$ with query complexity $q(n, \varepsilon, k) = \tilde{O}\big(1/\varepsilon^7\big)$, independent of k.*

In what follows, we assume that $k > 20/\varepsilon$, as otherwise we can use for instance the $O(k/\varepsilon)$-query (non-adaptive, one-sided) tester of Theorem 4 to obtain an $O\big(1/\varepsilon^2\big)$ query complexity.

3.1 Testing k-monotonicity over $[Ck]$

In this section, we give a $\operatorname{poly}(C/\varepsilon)$-query tester for k-monotonicity over the domain $[Ck]$, where C is a parameter to be chosen (for our applications, we will eventually set $C = \operatorname{poly}(1/\varepsilon)$).

▶ **Lemma 12.** *There exists a two-sided non-adaptive tester for k-monotonicity of functions $f\colon [Ck] \to \{0,1\}$ with query complexity $O\Big(\frac{C^3}{\varepsilon^3}\Big)$.*

The tester proceeds by reducing to support size estimation and using (a slight variant of) an algorithm of Canonne and Rubinfeld [13]. Let $f\colon [Ck] \to \{0,1\}$, and suppose f is s-monotone but not $(s-1)$-monotone. Then there is a unique partition of $[Ck]$ into $s+1$ disjoint intervals $I_1, I_2, \ldots, I_{s+1}$ such that f is constant on each interval; note that this constant value alternates in consecutive intervals. We can then define a distribution D_f over $[s+1]$ such that $D_f(i) = |I_i|/(Ck)$.

Our next claims, Claim 13 and Claim 14, provide the basis for the reduction (from testing k-monotonicity of f to support size estimation of D_f).

▶ **Claim 13.** *If f is ε-far from k-monotone, then it is not $(1+\frac{\varepsilon}{4})k$-monotone, and in particular $|\operatorname{supp}(D_f)| > (1+\frac{\varepsilon}{4})k + 1$.*

▶ **Claim 14.** *To ε-test k-monotonicity of f, it suffices to estimate $|\operatorname{supp}(D_f)|$ to within $\frac{\varepsilon k}{10}$.*

The proofs of above claims are relatively straightforward. So we defer these proofs to the end of this section and now proceed to use algorithm of [13] to do support size estimation. The algorithm of [13] uses "dual access" to D; an oracle that provides a random sample from D, and an oracle that given an element of D, returns the probability mass assigned to this element by D.

▶ **Theorem 15** ([13, Theorem 14 (rephrased)]). *In the access model described above, there exists an algorithm that, on input a threshold $n \in \mathbb{N}^*$ and a parameter $\varepsilon > 0$, and given access to a distribution D (over an arbitrary set) satisfying $\min_{x \in \mathrm{supp}(D)} D(x) \geq \frac{1}{n}$ estimates the support size $|\mathrm{supp}(D)|$ up to an additive εn, with query complexity $O\left(\frac{1}{\varepsilon^2}\right)$.*

Note however that we only have access to D_f through query access to f, and thus have to manage to simulate (efficiently) access to the former. One difficulty is that, to access $D_f(i)$, we need to determine where I_i lies in f. For example, finding $D_f(k/2)$ requires finding $I_{k/2}$, which might require a large number of queries to f. We circumvent this by weakening the "dual access" model in two ways, arguing for each of these two relaxations that the algorithm of [13] can still be applied:

- we rewrite the support size as in [13], as $|\mathrm{supp}(D_f)| = \mathbb{E}_{x \sim D_f}[1/D_f(x)]$. We want to estimate it to within $\pm O(\varepsilon k)$ which we can do by random sampling;
- the quantity inside the expectation depends on $D_f(x)$ but not x itself, so "labels" are unnecessary for our random sampling. Thus, it will be sufficient to be able to compute $D_f(x)$ (and thus $1/D_f(x)$) for a random $x \sim D_f$, even if not actually knowing x itself;
- actually, even calculating $D_f(x)$ may possibly too expensive, so instead we will estimate $\mathbb{E}_{x \sim D_f}[1/\tilde{D}_f(x)]$ where $\tilde{D}_f(x) = \min(\frac{20}{\varepsilon k}, D_f(x))$. Note that $\tilde{D}_f$ might no longer define a probability distribution; but this expectation is only off by at most $\frac{\varepsilon k}{20}$, since $1/D'_f(x) = \max(\varepsilon k/20, 1/D_f(x))$ and $1/D_f(x)$ is positive.

More details follow.

First, we note as discussed above that the algorithm does not require knowing the "label" of any element in the support of the distribution: the only access required is being able to randomly sample elements according to D_f, and evaluate the probability mass on the sampled points. This, in turn, can be done, as the following two lemmas explain:

▶ **Lemma 16** (Sampling from D_f). *Let $i \in [n]$ be chosen uniformly at random, and let j be such that $i \in I_j$. Then, the distribution of j is exactly D_f.*

▶ **Lemma 17** (Evaluating $D_f(j)$). *Suppose $I_j = \{a, a+1, \ldots, b\}$. Given i such that $i \in I_j$, we can find I_j by querying $f(i+1) = f(i+2) = \cdots = f(b)$ and $f(b+1) \neq f(b)$, as well as $f(i-1) = f(i-2) = \ldots = f(a)$ and $f(a-1) \neq f(a)$. The number of queries to f is $b - a + 3 = |I_j| + 3$.*

Here comes the second difficulty: if we straightforwardly use these approaches to emulate the required oracles to estimate the support size of D_f, the number of queries is potentially very large. For instance, if we attempt to query $D_f(j)$ where $|I_j| = \Omega(k)$, we will need $\Omega(k)$ queries to f. This is where comes the second relaxation: specifically, we shall argue that it will be enough for us to "cap" the size of the interval (as per our next lemma).

▶ **Lemma 18** (Evaluating $D_f(j)$ with a cap). *Given i such that $i \in I_j$, we will query f on every point in $[i - 20C/\varepsilon, i + 20C/\varepsilon]$. If $|I_j| \leq 20C/\varepsilon$, then I_j will be determined by these queries. If these queries do not determine I_j, we know $|I_j| > 20C/\varepsilon$. Beyond querying i, this requires $40C/\varepsilon$ (nonadaptive) queries.*

We now can put all the above pieces together and give the proof for Lemma 12:

Proof of Theorem 12: As previously discussed, we use the algorithm of [13] for estimating support size. Inspecting their algorithm, we see that our cap of $20C/\varepsilon$ for interval length (and therefore $20/(\varepsilon k)$ for maximum probability reported) might result in further error of the estimate. The algorithm interacts with the unknown function by estimating the expected value of $1/D_f(j)$ over random choices of j with respect to D_f. Our cap can only decrease this expectation by at most $(\varepsilon k)/20$. Indeed, the algorithm works by estimating the quantity $\mathbb{E}_{x\sim D_f}[\frac{1}{D_f(x)}\mathbb{1}_{\{D_f(x)>\tau\}}]$, for some suitable parameter $\tau > 0$. By capping the value of $1/D_f(x)$ to $20/(\varepsilon k)$, we can therefore only decrease the estimate, and by at most $20/(\varepsilon k) \cdot D_f(\{\ x :\ D_f(x) > (\varepsilon k)/20\ \}) \le 20/(\varepsilon k)$.

The condition for their algorithm to estimate support size to within $\pm\varepsilon m$ is that all elements in the support have a probability mass of at least $1/m$. Since each nonempty interval has length at least 1, we have $\min_j D_f(j) \ge (1/Ck)$. In order for their algorithm to report an estimate within $\pm\varepsilon k/20$ of support size, we set $\varepsilon' = (\varepsilon/20C)$ in their algorithm.

The total error in support size is at most $\varepsilon k/20 + \varepsilon k/20 = \varepsilon k/10$. By Claim 14, this suffices to test ε-test k-monotonicity of f.

Using the algorithm of [13], we need $O(1/{\varepsilon'}^2) = O\big((C/\varepsilon)^2\big)$ queries to D_f. For every query to D_f, we need to make $O(C/\varepsilon)$ queries to f, so the overall query complexity is $O\big(C^3/\varepsilon^3\big)$. ◀

Proof of Claim 13. The last part of the statement is immediate from the first, so it suffices to prove the first implication. We show the contrapositive: assuming f is $(1+\frac{\varepsilon}{4})k$-monotone, we will "fix" it into a k-monotone function by changing at most εn points. In what follows, we assume $\frac{\varepsilon k}{4} \ge 1$, as otherwise the statement is trivial (any function that is ε-far from k-monotone is *a fortiori* not k-monotone).

Let as before ℓ^* be the minimum integer ℓ for which f is ℓ-monotone: we can assume $k < \ell^* \le (1+\frac{\varepsilon}{4})k$ (as if $\ell^* \le k$ we are done.) Consider as above the maximal consecutive monochromatic intervals $I_1, \ldots, I_{\ell^*}$, and let i be the index of the shortest one. In particular, it must be the case that $|I_i| \le \frac{n}{\ell^*+1}$. Flipping the value of f on I_i therefore has "cost" at most $\frac{n}{\ell^*+1}$, and the resulting function f' is now exactly (ℓ^*-2)-monotone if $1 < i < \ell^*$, and (ℓ^*-1)-monotone if $i \in \{1, \ell^*\}$. This means in particular that repeating the above $\frac{\varepsilon}{4}k$ times is enough to obtain a k-monotone function, and the total cost is upperbounded by

$$\sum_{j=0}^{\varepsilon k/4} \frac{n}{\ell^*+1-2j} \le \sum_{j=0}^{\varepsilon k/4} \frac{n}{k+1-2j} = \sum_{j=k(1-\frac{\varepsilon}{2})+1}^{k+1} \frac{n}{j} \le n\frac{\frac{\varepsilon}{2}k+1}{(1-\frac{\varepsilon}{4})k+1} \le n\frac{\frac{3\varepsilon}{4}k}{(1-\frac{\varepsilon}{4})k}$$

where for the last inequality (for the numerator) we used that $1 \le \frac{\varepsilon k}{4}$. But this last RHS is upperbounded by εn (as $\frac{3}{4}x \le x(1-\frac{1}{4}x)$ for $x \in [0,1]$), showing that therefore, f was ε-close to k-monotone to begin with, which is a contradiction. ◀

Proof of Claim 14. If f is ε-far from k-monotone, then $|\mathrm{supp}(D_f)| > (1+\frac{\varepsilon}{4})k = k + \frac{\varepsilon}{4}k$, and if f is k-monotone, then $|\mathrm{supp}(D_f)| \le k+1$. The fact that $k > 20/\varepsilon$ then allows us to conclude. ◀

3.2 Reducing $[n] \to \{0,1\}$ to $[Ck] \to \{0,1\}$.

Now we show how to reduce ε-testing k-monotonicity of $f\colon [n] \to \{0,1\}$ to ε'-testing k-monotonicity of a function $g\colon [Ck] \to \{0,1\}$ for $C = \mathrm{poly}(1/\varepsilon)$ and $\varepsilon' = \mathrm{poly}(\varepsilon)$, resulting in a $\mathrm{poly}(1/\varepsilon)$-query algorithm for ε-testing k-monotonicity.

The first step is (as before) to divide $[n]$ into blocks (disjoint intervals) of size $\frac{\varepsilon n}{4k}$ if $\varepsilon > \frac{8k}{n}$ (again assuming without loss of generality that $\frac{\varepsilon n}{4k}$ is an integer), and blocks of size 1 otherwise (in which case $n \leq \frac{8k}{\varepsilon}$ and we can directly apply the result of Theorem 12, with $C = n/k \leq 8/\varepsilon$). Let $m = 4k/\varepsilon$ be the number of resulting blocks, and define $f_m \colon [n] \to \{0,1\}$ as the *m-block-coarsening* of f: namely, for any $j \in B_i$, we set

$$f_m(j) = \text{argmax}_{b \in \{0,1\}} \Pr_{k \in B_i} [f_m(k) = b] \qquad \text{(majority vote)}$$

Ordering the blocks $B_1, B_2, \ldots, B_m$, we also define $g \colon [m] \to \{0,1\}$ such that $g(i) = \min_{a \in B_i} f_m(a)$.

It is easy to see that if f is k-monotone, then f has at most k non-constant blocks, and f_m is k-monotone. Because the function f only changes values k times; for a block to be non-constant, the block must contain a pair of points with a value change. We call a block *variable* if the minority points comprise at least an $\varepsilon/100$-fraction of the block; formally, B is variable if $\min_{b \in \{0,1\}} \Pr_{j \in B}[f(j) = b] \geq \varepsilon/100$.

We need following claims (their proofs are at the end of the section) to prove Theorem 5.

▶ **Claim 19.** *Suppose f has s variable blocks. Then* $\text{dist}(f, f_m) \leq s/m + \varepsilon/100$.

▶ **Claim 20.** *Suppose f is promised to be either (i) k-monotone or (ii) such that f_m has more than $\frac{5}{4}k$ variable blocks. Then we can determine which with $O\big(\frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon}\big)$ queries, and probability* 9/10.

Proof of Theorem 5. We use the estimation/test from the previous claim as the first part of our tester. Assuming f passes, we can assume that f_m has less than $\frac{5}{4}k$ variable blocks. By Claim 19, $\text{dist}(f, f_m) \leq \frac{5k}{4}/m + \frac{\varepsilon}{100} = \frac{5\varepsilon}{32} + \frac{\varepsilon}{100} \leq \frac{\varepsilon}{3}$. This part takes $O\big(\frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon}\big)$ queries.

Now, we apply the tester of Theorem 12 (with probability of success amplified to 9/10 by standard arguments) to $(\varepsilon/6)$-test k-monotonicity of $g \colon [m] \to \{0,1\}$, where $g(i)$ is the constant value of f_m on B_i, and $m = (4k)/\varepsilon$. Let q be the query complexity of the tester, and set $\delta = 1/(10q)$; to query $g(i)$, we randomly query f on $O\big(\frac{1}{\varepsilon} \log \frac{1}{\delta}\big)$ points in B_i and take the majority vote. With probability at least $1 - \delta$, we get the correct value of $g(i)$, and by a union bound all q simulated queries have the correct value with probability at least 9/10.

Therefore, to get a single query to g, we use $O((\log q)/\varepsilon)$ queries. In the context of our previous section, we have $C = 4/\varepsilon$, so $q = O(C^3/\varepsilon^3) = O\big(1/\varepsilon^6\big)$ and the overall query complexity of this part is $O((q \log q)/\varepsilon) = O\big(\frac{1}{\varepsilon^7} \log \frac{1}{\varepsilon}\big)$. This dominates the query complexity of the other part of the tester, from Claim 20, which is $O\big(\frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon}\big)$. By a union bound over the part from Claim 20, the simulation of g, and the call to the tester of Theorem 12, the algorithm is correct with probability at least $1 - 3/10 > 2/3$. ◀

Proof of Claim 19. We will estimate the error of f_m in computing f on variable blocks and non-variable blocks separately. Each non-variable block B can contribute error on at most $\varepsilon |B| /100$ points. Each variable block B can contribute error on at most $|B| = n/m$ points. The total number of errors is at most $\varepsilon n/100 + s(n/m) = n(\varepsilon/100 + s/m)$, yielding the upper bound on $\text{dist}(f, f_m)$. ◀

Proof of Claim 20. We first note that given any fixed block B, it is easy to detect whether it is variable (with probability of failure at most δ) by making $O\big(\frac{1}{\varepsilon} \log \frac{1}{\delta}\big)$ uniformly distributed queries in B. Doing so, a variable block will be labelled as such with probability at least $1 - \delta$, while a constant block will never be marked as variable. (If a block is neither constant nor variable, then any answer will do.)

Letting s denote the number of variable blocks, we then want to non-adaptively distinguish between $s \geq \frac{5}{4}k = \frac{5\varepsilon}{16}m$ and $s \leq k = \frac{\varepsilon}{4}m$ (since if f were k-monotone, then f_m had at most k variable blocks). Doing so with probability at least 19/20 can be done by checking only $q = O\left(\frac{1}{\varepsilon}\right)$ blocks chosen uniformly at random: by the above, setting $\delta = \frac{1}{20q}$ all of the q checks will also yield the correct answer with probability no less than 9/10, so by a union bound we will distinguish (i) and (ii) with probability at least 9/10. We conclude by observing that all $O\left(q \cdot \frac{1}{\varepsilon}\log\frac{1}{q}\right) = O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon}\right)$ queries are indeed non-adaptive. ◀

4 On the high-dimensional grid

In this section, we give two algorithms for tolerant testing, that is testing whether a function $f\colon [n]^d \to \{0,1\}$ is ε_1-close to k-monotone vs. ε_2-far from k-monotone, establishing Theorem 8. The first has query complexity exponential in the dimension d and is *fully tolerant*, that is works for any setting of $0 \leq \varepsilon_1 < \varepsilon_2 \leq 1$. The second applies whenever $\varepsilon_2 > 3\varepsilon_1$, and has (incomparable) query complexity exponential in $\tilde{O}(k\sqrt{d}/(\varepsilon_2 - 3\varepsilon_1)^2)$. Both of these algorithms can be used for non-tolerant ("regular") testing by setting $\varepsilon_1 = 0$ and $\varepsilon_2 = \varepsilon$, which implies Theorem 7.

▶ **Theorem 8.** *There exist*

- *a non-adaptive (fully) tolerant tester for k-monotonicity of functions $f\colon [n]^d \to \{0,1\}$ with query complexity $q(n,d,\varepsilon_1,\varepsilon_2,k) = \tilde{O}\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\left(\frac{5kd}{\varepsilon_2-\varepsilon_1}\right)^d\right)$;*
- *a non-adaptive tolerant tester for k-monotonicity of functions $f\colon [n]^d \to \{0,1\}$ with query complexity $q(n,d,\varepsilon_1,\varepsilon_2,k) = 2^{\tilde{O}\left(k\sqrt{d}/(\varepsilon_2-3\varepsilon_1)^2\right)}$, under the restriction that $\varepsilon_2 > 3\varepsilon_1$.*

As a corollary, this implies Theorem 7, restated below:

▶ **Theorem 7.** *There exists a non-adaptive tester for k-monotonicity of functions $f\colon [n]^d \to \{0,1\}$ with query complexity $q(n,d,\varepsilon,k) = \min(\tilde{O}\left(\frac{1}{\varepsilon^2}\left(\frac{5kd}{\varepsilon}\right)^d\right), 2^{\tilde{O}\left(k\sqrt{d}/\varepsilon^2\right)})$.*

For convenience, we will view in this part of the paper the set $[n]$ as $[n] = \{0,1,\ldots,n-1\}$. Assuming that m divides n, we let $\mathcal{B}_{m,n}\colon [n]^d \to [m]^d$ be the mapping such that $\mathcal{B}_{m,n}(y)_i = \lfloor y_i/m \rfloor$ for $1 \leq i \leq m$. For $x \in [m]^d$, we define the set $\mathcal{B}^{-1}_{m,n}(x)$ to be the inverse image of x. Specifically, $\mathcal{B}^{-1}_{m,n}(x)$ is the set of points of the form $m \cdot x + [n/m]^d$, with standard definitions for scalar multiplication and coordinate-wise addition. That is, $\mathcal{B}^{-1}_{m,n}(x)$ is a "coset" of $[n/m]^d$ points in $[n]^d$. To keep on with the notations of the other sections, we will call these cosets *blocks*, and will say a function $h\colon [n]^d \to \{0,1\}$ is an *m-block function* if it is constant on each block. Moreover, for clarity of presentation, we will omit the subscripts on $\mathcal{B}$ and $\mathcal{B}^{-1}$ whenever they are not necessary.

We first establish a lemma that will be useful for the proofs of correctness of both algorithms.

▶ **Lemma 21.** *Suppose $f\colon [n]^d \to \{0,1\}$ is k-monotone. Then there is an m-block function $h\colon [n]^d \to \{0,1\}$ such that $\mathrm{dist}(f,h) < kd/m$.*

Proof. Fix any k-monotone function $f\colon [n]^d \to \{0,1\}$. We partition $[m]^d$ into chains of the form

$$C_x = \left\{ x + \ell \cdot 1^d \,:\, \ell \in \mathbb{N}, x \in [m]^d \text{ and } x_i = 0 \text{ for some } i \right\}.$$

There are $m^d - (m-1)^d \leq dm^{d-1}$ of these chains: we will show that f can only be nonconstant on at most k blocks of each chain.

Algorithm 1 Fully tolerant testing with $O(kd/(\varepsilon_2-\varepsilon_1)))^d$ queries.

Require: Query access to $f\colon [n]^d \to \{0,1\}$, $\varepsilon_2 > \varepsilon_1 \geq 0$, a positive integer k

$\alpha \leftarrow (\varepsilon_2 - \varepsilon_1), m \leftarrow \lceil 5kd/\alpha \rceil, t \leftarrow \lceil 25\ln(6m^d)/(2\alpha^2) \rceil$

▷ Define a distribution D over $[m]^d \times \{0,1\}$.

for $x \in [m]^d$ **do**

  Query f on t random points $T_x \subseteq \mathcal{B}^{-1}(x)$.

  $D(x,0) \leftarrow \Pr_{y\in T_x}[\,f(y)=0\,]/m^d$

  $D(x,1) \leftarrow \Pr_{y\in T_x}[\,f(y)=1\,]/m^d$

end for

▷ Define a distribution D' over $[n]^d \times \{0,1\}$ such that $D'(y,b) = D(\mathcal{B}(y),b)\cdot m^d/n^d$.

if there exists a k-monotone m-block function h such that $\Pr_{(y,b)\sim D'}[\,h(y)\neq b\,] \leq \varepsilon_1 + \frac{\alpha}{2}$ **then return** ACCEPT

end if

return REJECT

By contradiction, suppose there exists $x \in [m]^d$ such that f is nonconstant on $k+1$ different blocks $\mathcal{B}^{-1}(z^{(i)})$, where $z^{(1)} \prec z^{(2)} \prec \ldots \prec z^{(k)} \prec z^{(k+1)}$, and each $z^{(i)} \in C_x$. By construction, we have $\mathcal{B}^{-1}(z^{(i)}) \prec \mathcal{B}^{-1}(z^{(j)})$ for $i<j$. For each $1 \leq i \leq k+1$, there are two points $v_*^{(i)}, v_{**}^{(i)} \in \mathcal{B}^{-1}(z_i)$ such that $v_*^{(i)} \prec v_{**}^{(i)}$ and $f(v_*^{(i)}) \neq f(v_{**}^{(i)})$. By construction $v_*^{(1)} \prec v_{**}^{(1)} \prec v_*^{(2)} \prec v_{**}^{(2)} \prec v_*^{(3)} \prec v_{**}^{(3)} \prec \ldots \prec v_*^{(k+1)} \prec v_{**}^{(k+1)}$, and there must be at least $k+1$ pairs of consecutive points with differing function values. Out of these $2k+2$ many points, there is a chain of points $\bar{v}^{(1)} \prec \bar{v}^{(2)} \prec \ldots \prec \bar{v}^{(k+1)}$ where $f(\bar{v}^{(i)}) \neq f(\bar{v}^{(i+1)})$ for $1 \leq i \leq k$, which is a violation of the k-monotonicity of f.

Thus, in each of the dm^{d-1} many chains of blocks, there can only be k nonconstant blocks. It follows that there are at most kdm^{d-1} nonconstant blocks in total. We now define $h(y)$ to be equal to $f(y)$ if f is constant on $\mathcal{B}(y)$, and arbitrarily set $h(y)=0$ otherwise. Each set $\mathcal{B}^{-1}(y)$ contains $(n/m)^d = n^d \cdot m^{-d}$ many points, and f is not constant on at most kdm^{d-1} of these. It follows that $\mathrm{dist}(f,h) \leq kdm^{d-1}\cdot m^{-d} = kd/m$. ◀

4.1 Fully tolerant testing with $O(kd/(\varepsilon_2-\varepsilon_1)))^d$ queries

Our first algorithm (Algorithm 1) then proceeds by essentially brute-force learning an m-block function close to the unknown function, and establishes the first item of Theorem 8.

▶ **Proposition 22.** *Algorithm 1 accepts all functions ε_1-close to k-monotone functions, and rejects all functions ε_2-far from k-monotone (with probability at least $2/3$). Its query complexity is $O\left(\frac{d}{(\varepsilon_2-\varepsilon_1)^2}\left(\frac{5kd}{\varepsilon_2-\varepsilon_1}+1\right)^d \log\frac{kd}{\varepsilon_2-\varepsilon_1}\right)$.*

Proof. The algorithm first estimates $\Pr_{y\in\mathcal{B}^{-1}(x)}[\,f(y)=b\,]$ for every $x \in [m]^d$ and $b \in \{0,1\}$ to within $\pm\frac{\alpha}{5}$. We use $t = 25\ln(6m^d)/2\alpha^2$ points in each block to ensure (by an additive Chernoff bound) that each estimate is correct except with probability at most $m^{-d}/3$. By a union bound, the probability that all estimates are correct is at least $2/3$, and we hereafter condition on this. By construction, $\mathbb{E}_{(x,b)\sim D}[\Pr_{y\in\mathcal{B}^{-1}(x)}[\,f(y)\neq b\,]] = \Pr_{(y,b)\sim D'}[\,f(y)\neq b\,] \leq \frac{\alpha}{5}$. In this probability experiment, the marginal distribution of D' on y is uniform over $[n]^d$.

Let $f^*\colon [n]^d \to \{0,1\}$ be a k-monotone function minimizing $\Pr[\,f(y)\neq f^*(y)\,]$. Theorem 21 ensures that there is a k-monotone m-block function $h\colon [n]^d \to \{0,1\}$ such that $\mathrm{dist}(f^*,h) < kd/m \leq \alpha/5$. Let $h^*\colon [n]^d \to \{0,1\}$ be a k-monotone m-block function minimizing $\mathrm{dist}(f^*,h^*)$.

Completeness

Suppose $\mathrm{dist}(f, f^*) \le \varepsilon_1$. Then by the triangle inequality,

$$\begin{aligned}\Pr_{(y,b)\sim D'}[\,h^*(y) \neq b\,] &\le \Pr_{(y,b)\sim D'}[\,h^*(y) \neq f^*(y)\,] + \Pr_{(y,b)\sim D'}[\,f^*(y) \neq f(y)\,] \\ &\quad + \Pr_{(y,b)\sim D'}[\,f(y) \neq b\,] \\ &\le \varepsilon_1 + \frac{2\alpha}{5}.\end{aligned}$$

where to bound the first term $\Pr_{(y,b)\sim D'}[\,h^*(y) \neq f^*(y)\,]$ by $\mathrm{dist}(f^*, h^*) \le \alpha/5$ we used the fact that the marginal distribution of y is uniform when $(y, b) \sim D'$. Thus, the algorithm will find a k-monotone m-block function close to D (without using any queries to f) and accept.

Soundness

Suppose $\mathrm{dist}(f, f^*) \ge \varepsilon_2$. Then by the triangle inequality

$$\begin{aligned}\Pr_{(y,b)\sim D'}[h(y) \neq b] &\ge \Pr_{(y,b)\sim D'}[h(y) \neq f(y)] - \Pr_{(y,b)\sim D'}[f(y) \neq b] \\ &\ge \Pr_{(y,b)\sim D'}[f^*(y) \neq f(y)] - \Pr_{(y,b)\sim D'}[f(y) \neq b] \\ &\ge \varepsilon_2 - \frac{\alpha}{5}\end{aligned}$$

for every k-monotone m-block function h. Since $\varepsilon_2 - 2\alpha/5 \ge \varepsilon_1 + 3\alpha/5$, the algorithm never find a k-monotone m-block function h with low error with respect to D, and the algorithm will reject.

Query complexity

The algorithm only makes queries in constructing D; the number of queries required is $m^d \cdot t = O\Big(\frac{d}{\alpha^2}\left(\frac{5kd}{\alpha} + 1\right)^d \log\frac{kd}{\alpha}\Big)$. ◀

4.2 Tolerant testing *via* agnostic learning

We now present our second algorithm, Algorithm 2, proving the second item of Theorem 8. At its core is the use of an *agnostic learning algorithm* for k-monotone functions, which we first describe.[2]

▶ **Proposition 23.** *There exists an agnostic learning algorithm for k-monotone functions over $[r]^d \to \{0,1\}$ with excess error τ with sample complexity* $\exp(\widetilde{O}(k\sqrt{d}/\tau^2))$.

We will rely on tools from Fourier analysis to prove Proposition 23. For this reason, it will be convenient in this section to view the range as $\{-1, 1\}$ instead of $\{0, 1\}$.

[2] Recall that an *agnostic learner with excess error τ* for some class of functions $\mathcal{C}$ is an algorithm that, given an unknown distribution D, an unknown arbitrary function f, and access to random labelled samples $\langle x, f(x)\rangle$ where $x \sim D$, satisfies the following. It outputs a hypothesis function $\hat{h}$ such that $\Pr_{x\sim D}\left[\,f(x) \neq \hat{h}(x)\,\right] \le \mathrm{OPT}_D + \tau$ with probability at least $2/3$, where $\mathrm{OPT}_D = \min_{h\in\mathcal{C}} \Pr_{x\sim D}[\,f(x) \neq h(x)\,]$ (i.e., it performs "almost as well as the best function in $\mathcal{C}$").

Algorithm 2 Multiplicative approximation with $\exp(\widetilde{O}(k\sqrt{d}/((\varepsilon_2-3\varepsilon_1)^2))$ queries.

Require: Query access to $f\colon [n]^d \to \{0,1\}$, $\varepsilon_2 > 3\varepsilon_1 \geq 0$, a positive integer k

1: $\alpha \leftarrow (\varepsilon_2 - 3\varepsilon_1)$, $m \leftarrow \lceil 6kd/\varepsilon \rceil$, $t \leftarrow \lceil 3d(k+1)/\varepsilon \ln m + \ln 100 \rceil$
2: Define D to be the distribution over $[m]^d \times \{0,1\}$ such that $D(x,b) = \Pr_{y\in\mathcal{B}^{-1}(x)}[\, f(y) = b\,]$.
3: ▷ $\mathcal{A}_D(\tau, f)$ denotes the output of an agnostic learner of k-monotone functions with respect to D, with excess error τ and probability of failure $1/10$
4: $h\colon [m]^d \to \mathbb{R} \leftarrow \mathcal{A}_D(\alpha/12, f)$.
5: Estimate $\Pr_{(x,b)\sim D}[\, h(x) \neq b\,]$ to within $\pm\alpha/7$ with probability of failure $1/10$, using $O(1/\alpha^2)$ queries.
6: **if** the estimate is more than $\varepsilon_1 + \frac{5\alpha}{12}$ **then return** REJECT
7: **end if**
8: **if** $\mathrm{dist}(h,\ell) = \Pr_{x\in[m]^d}[\, h(x) \neq \ell(x)\,] \leq 2\varepsilon_1 + \frac{5\alpha}{12}$ for some k-monotone m-block function ℓ **then return** ACCEPT
9: **else return** REJECT
10: **end if**

▶ **Definition 24.** For a Boolean function $f\colon [r]^d \to \{-1,1\}$, we define

$$\mathbf{Inf}_i[f] = 2\Pr\Big[\, f(x) \neq f(x^{(i)})\,\Big]$$

where $x = (x_1, x_2, \ldots, x_d)$ is a uniformly random string over $[r]^d$, and

$$x^{(i)} = (x_1, x_2, \ldots, x_{i-1}, x_i', x_{i+1}, \ldots, x_d)$$

for x' drawn independently and uniformly from $[r]$. We also define $\mathbf{Inf}[f] = \sum_{i=1}^d \mathbf{Inf}_i[f]$.

We first generalize the following result, due to Blais *et al.*, for more general domains:

▶ **Proposition 25** ([9]). *Let $f\colon \{0,1\}^d \to \{-1,1\}$ be a k-monotone function. Then $\mathbf{Inf}[f] \leq k\sqrt{d}$.*

▶ **Lemma 26** (Generalization). *Let $f\colon [r]^d \to \{-1,1\}$ be a k-monotone function. Then $\mathbf{Inf}[f] \leq k\sqrt{d}$.*

Proof. For any two strings $y^0, y^1 \in [r]^d$, let $f_{y^0,y^1}\colon \{0,1\}^d \to \{-1,1\}$ be the function obtained by setting $f_{y^0,y^1}(x) = f(y^x)$, where $y^x \in [r]^d$ is defined as

$$y_i^x = \begin{cases} \min\{y_i^0, y_i^1\} & \text{if } x_i = 0 \\ \max\{y_i^0, y_i^1\} & \text{if } x_i = 1 \end{cases}$$

Since f was a k-monotone function, so is f_{y^0,y^1}. Thus $\mathbf{Inf}[f_{y^0,y^1}] \leq k\sqrt{d}$ for every choice of y^0 and y^1. It is not hard to see that for any fixed $i \in [d]$ the following two processes yield the same distribution over $[r]^d \times [r]^d$:

- Draw $z \in [r]^d$, $z_i' \in [r]$ independently and uniformly at random, set

$$z' \stackrel{\text{def}}{=} (z_1, \ldots, z_{i-1}, z_i', z_{i+1}, \ldots, z_d),$$

  and output (z, z');
- Draw $y^0, y^1 \in [r]^d, x \in \{0,1\}^d$ independently and uniformly at random, and output $(y^x, y^{x^{(i)}})$.

This implies that

$$\begin{aligned}
\mathbf{Inf}[f] &= \sum_{i=1}^{d} \mathbf{Inf}_i[f] = \sum_{i=1}^{d} 2 \Pr_{z\in[r]^d}[f(z) \neq f(z^{(i)})] \\
&= \sum_{i=1}^{d} 2\mathbb{E}_{y^0,y^1\in[r]^d}\left[\Pr_{x\in\{0,1\}^d}\left[f(y^x) \neq f(y^{x^{(i)}})\right]\right] \\
&= \mathbb{E}_{y^0,y^1\in[r]^d}\left[\sum_{i=1}^{d} 2 \Pr_{x\in\{0,1\}^d}\left[f(y^x) \neq f(y^{x^{(i)}})\right]\right] \\
&= \mathbb{E}_{y^0,y^1\in[r]^d}\left[\sum_{i=1}^{d} 2 \Pr_{x\in\{0,1\}^d}\left[f_{y_0,y_1}(x) \neq f_{y_0,y_1}(x^{(i)})\right]\right] \\
&= \mathbb{E}_{y^0,y^1}[\mathbf{Inf}[f_{y^0,y^1}]] \leq \mathbb{E}_{y^0,y^1}[k\sqrt{d}] = k\sqrt{d}.
\end{aligned}$$

◀

For two functions $f, g\colon [r]^d \to \mathbb{R}$, we define the inner product $\langle f, g\rangle = \mathbb{E}_x[f(x)g(x)]$, where the expectation is taken with respect to the uniform distribution. It is known that for functions $f\colon [r]^d \to \mathbb{R}$, there is a "Fourier basis" of orthonormal functions f. To construct such a basis, we can take any orthonormal basis $\{\phi_0 \equiv 1, \phi_1, \ldots, \phi_{|r|-1}\}$ for functions $f\colon [r] \to \mathbb{R}$. Given such a basis, a Fourier basis is the collection of functions ϕ_α, where $\alpha \in [r]^d$, and $\phi_\alpha(x) = \prod_{i=1}^{d} \phi_{\alpha_i}(x_i)$. Then every $f\colon [r]^d \to \mathbb{R}$ has a unique representation $f = \sum_{\alpha\in[r]^d} \widehat{f}(\alpha)\phi_\alpha$, where $\widehat{f}(\alpha) = \langle f, \phi_\alpha\rangle \in \mathbb{R}$.

Many Fourier formulæ hold in arbitrary Fourier bases, an important example being Parseval's Identity: $\sum_{\alpha\in[r]^d} \widehat{f}(\alpha)^2 = 1$. We will use the following property:

▶ **Lemma 27** ([39, Proposition 8.23]). *For $\alpha \in [r]^d$, let $|\alpha|$ denote the number of nonzero coordinates in α. Then we have*

$$\mathbf{Inf}[f] = \sum_{\alpha\in[r]^d} |\alpha|\, \hat{f}(\alpha)^2.$$

▶ **Lemma 28.** *If $\mathbf{Inf}[f] \leq k$, then $\sum_{\alpha:|\alpha|>k/\varepsilon} \widehat{f}(\alpha)^2 \leq \varepsilon$.*

Proof. If not, then $\mathbf{Inf}[f] = \sum_\alpha |\alpha|\, \widehat{f}(\alpha)^2 \geq \sum_{\alpha:|\alpha|>k/\varepsilon} |\alpha|\, \widehat{f}(\alpha)^2 \geq \frac{k}{\varepsilon} \sum_{\alpha:|\alpha|>k/\varepsilon} \widehat{f}(\alpha)^2 > \frac{k}{\varepsilon} \cdot \varepsilon = k$, a contradiction. ◀

▶ **Lemma 29.** *Let p be the function $\sum_{\alpha:|\alpha|\leq t} \widehat{f}(\alpha)\phi_\alpha$. Then*

- **(i)** $\|p - f\|_2^2 = \mathbb{E}_{x\in[r]^d}[(p(x) - f(x))^2] = \sum_{\alpha:|\alpha|>t} \widehat{f}(\alpha)^2$;
- **(ii)** *p is expressible as a linear combination of real-valued functions over $[r]^d$, each of which only depends on at most t coordinates;*
- **(iii)** *p is expressible as a degree-t polynomial over the rd indicator functions $\mathbb{1}_{\{x_i=j\}}$ for $1 \leq i \leq d$ and $j \in [r]$.*

▶ **Theorem 30** ([30, Theorem 5]). *Let $\mathcal{C}$ be a class of Boolean functions over $\mathcal{X}$ and $\mathcal{S}$ a collection of real-valued functions over $\mathcal{X}$ such that for every $f\colon \mathcal{X} \to \{-1, 1\}$ in $\mathcal{C}$, there exists a function $p\colon \mathcal{X} \to \mathbb{R}$ such that p is expressible as a linear combination of functions from $\mathcal{S}$ and $\|p - f\|_2^2 \leq \tau^2$. Then there is an agnostic learning algorithm for $\mathcal{C}$ achieving excess error τ which has sample complexity* $\mathrm{poly}(|\mathcal{S}|, 1/\tau)$.

Importantly, this algorithm is still successful with inconsistent labelled samples (examples), as long as they come from a distribution on $\mathcal{X} \times \{-1, 1\}$, where the marginal distribution on $\mathcal{X}$ is uniform.

Now we put all the pieces together. To agnostically learn a k-monotone function, we simply perform the agnostic learning algorithm of [30] on the distribution D over $[m]^d \times \{-1, 1\}$ defined by

$$D(x, b) = \Pr_{y \in \mathcal{B}^{-1}(x)} [\, f(y) = b \,] .$$

To generate a sample (x, b) from D, we draw a uniformly random string in $x \in [m]^d$, and b is the result of a query for the value of $f(y)$ for a uniformly random $y \in \mathcal{B}^{-1}(x)$. From Theorem 29, we can take $\mathcal{S}$ to be the set of $(k\sqrt{d}/\tau^2)$-way products of rd indicator functions. It follows that $|\mathcal{S}| = \binom{rd}{k\sqrt{d}/\tau^2} = \exp(\widetilde{O}(k\sqrt{d}/\tau^2))$.

▶ **Proposition 31.** *Algorithm 2 accepts all functions ε_1-close to k-monotone functions, and rejects all functions ε_2-far from k-monotone, when $\varepsilon_2 > 3\varepsilon_1$ (with probability at least $2/3$). Its query complexity is $\exp(\widetilde{O}(k\sqrt{d}/(\varepsilon_2 - 3\varepsilon_1)^2))$.*

Proof. By a union bound, we have that with probability at least $8/10$ both Step 5 and Step 4 succeed. We hereafter condition on this.

Completeness

Suppose f is ε_1-close to k-monotone. Theorem 21 and the triangle inequality imply that there is a k-monotone m-block function g^* such that $\mathrm{dist}(f, g^*) \leq \varepsilon_1 + \alpha/6$. The agnostic learning algorithm thus returns a hypothesis h such that $\mathrm{dist}(f, h) \leq \varepsilon_1 + \alpha/4$. The algorithm estimates this closeness to within $\alpha/7$, so the estimate obtained in Step 5 is at most $\varepsilon_1 + \varepsilon/4 + \varepsilon/7 < \varepsilon_1 + 5\alpha/12$ and the algorithm does not reject in this step. By the triangle inequality, h is $(2\varepsilon_1 + 5\alpha/12)$-close to k-monotone, and the algorithm will accept. There is no estimation error here, since no queries to f are required.

Soundness

Now suppose f is ε_2-far from k-monotone, where $\varepsilon_2 = 3\varepsilon_1 + \alpha$ for some $\alpha > 0$. Suppose the algorithm does not reject when estimating $\mathrm{dist}(f, h)$, where h is the hypothesis returned by the agnostic learning algorithm. Then $\mathrm{dist}(f, h) \leq \varepsilon_1 + 5\alpha/12 + \alpha/7 < \varepsilon_1 + 7\alpha/12$. By the triangle inequality, if t is a k-monotone function, $\mathrm{dist}(h, t) \geq \mathrm{dist}(f, t) - \mathrm{dist}(f, h) > \varepsilon_2 - (\varepsilon_1 + 7\alpha/12) = 2\varepsilon_1 + 5\alpha/12$. The algorithm will thus reject in the final step.

Query complexity

The query complexity of the algorithm is dominated by the query complexity of the agnostic learning algorithm, which is $\exp(\widetilde{O}(k\sqrt{d}/\alpha^2)) = \exp(\tilde{O}\Big(k\sqrt{d}/(\varepsilon_2 - 3\varepsilon_1)^2\Big))$. ◀

Acknowledgments. We would like to thank Eric Blais for helpful remarks on an earlier version of this paper, and an anonymous reviewer for very detailed and insightful comments.

References

1 Nir Ailon and Bernard Chazelle. Information theory in property testing and monotonicity testing in higher dimension. *Inf. Comput.*, 204(11):1704–1717, 2006.

2 Kazuyuki Amano and Akira Maruoka. A superpolynomial lower bound for a circuit computing the Clique function with at most $(1/6) \log \log n$ negation gates. *SIAM Journal on Computing*, 35(1):201–216, 2005.

3 Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1987.

4 Maria-Florina Balcan, Eric Blais, Avrim Blum, and Liu Yang. Active property testing. In *FOCS*, pages 21–30. IEEE Computer Society, 2012.

5 Tŭgkan Batu, Ronitt Rubinfeld, and Patrick White. Fast approximate PCPs for multidimensional bin-packing problems. *Inf. Comput.*, 196(1):42–56, 2005.

6 Aleksandrs Belovs and Eric Blais. A polynomial lower bound for testing monotonicity. In *STOC*, pages 1021–1032. ACM, 2016.

7 Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. L_p-testing. In *STOC*, pages 164–173. ACM, 2014.

8 Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. *Computational Complexity*, 21(2):311–358, 2012.

9 Eric Blais, Clément L. Canonne, Igor Carboni Oliveira, Rocco A. Servedio, and Li-Yang Tan. Learning circuits with few negations. In *APPROX-RANDOM*, volume 40 of *LIPIcs*, pages 512–527. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

10 Jop Briët, Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Monotonicity testing and shortest-path routing on the cube. *Combinatorica*, 32(1):35–53, 2012.

11 Nader H. Bshouty and Christino Tamon. On the Fourier spectrum of monotone functions. *J. ACM*, 43(4):747–770, 1996.

12 Clément L. Canonne, Elena Grigorescu, Siyao Guo, Akash Kumar, and Karl Wimmer. Testing k-monotonicity. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:136, 2016.

13 Clément L. Canonne and Ronitt Rubinfeld. Testing probability distributions underlying aggregated data. In *ICALP (1)*, volume 8572 of *Lecture Notes in Computer Science*, pages 283–295. Springer, 2014.

14 Deeparnab Chakrabarty and C. Seshadhri. An $o(n)$ monotonicity tester for boolean functions over the hypercube. In *STOC*, pages 411–418. ACM, 2013. Journal version as [17].

15 Deeparnab Chakrabarty and C. Seshadhri. Optimal bounds for monotonicity and Lipschitz testing over hypercubes and hypergrids. In *STOC*, pages 419–428, 2013.

16 Deeparnab Chakrabarty and C. Seshadhri. An optimal lower bound for monotonicity testing over hypergrids. *Theory of Computing*, 10:453–464, 2014.

17 Deeparnab Chakrabarty and C. Seshadhri. An $o(n)$ Monotonicity Tester for Boolean Functions over the Hypercube. *SIAM J. Comput.*, 45(2):461–472, 2016.

18 Xi Chen, Anindya De, Rocco A. Servedio, and Li-Yang Tan. Boolean function monotonicity testing requires (almost) $n^{1/2}$ non-adaptive queries. In *STOC*, pages 519–528. ACM, 2015.

19 Xi Chen, Rocco A. Servedio, and Li-Yang Tan. New algorithms and lower bounds for monotonicity testing. In *FOCS*, pages 286–295. IEEE Computer Society, 2014.

20 Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. In *RANDOM-APPROX*, volume 1671 of *Lecture Notes in Computer Science*, pages 97–108. Springer, 1999.

21 Funda Ergün, Sampath Kannan, Ravi Kumar, Ronitt Rubinfeld, and Mahesh Viswanathan. Spot-checkers. *J. Comput. Syst. Sci.*, 60(3):717–751, 2000.

22 Shahar Fattal and Dana Ron. Approximating the distance to monotonicity in high dimensions. *ACM Trans. Algorithms*, 6(3), 2010.

23 Eldar Fischer. On the strength of comparisons in property testing. *Inf. Comput.*, 189(1):107–116, 2004.

24 Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 474–483, 2002.

25 Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samorodnitsky. Testing monotonicity. *Combinatorica*, 20(3):301–337, 2000.

26 Siyao Guo and Ilan Komargodski. Negation-limited formulas. In *APPROX-RANDOM*, volume 40 of *LIPIcs*, pages 850–866. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

27 Siyao Guo, Tal Malkin, Igor Carboni Oliveira, and Alon Rosen. The power of negations in cryptography. In *TCC (1)*, volume 9014 of *Lecture Notes in Computer Science*, pages 36–65. Springer, 2015.

28 Shirley Halevy and Eyal Kushilevitz. Testing monotonicity over graph products. *Random Struct. Algorithms*, 33(1):44–67, 2008.

29 Stasys Jukna. *Boolean Function Complexity*. Springer, 2012.

30 Adam Tauman Kalai, Adam R. Klivans, Yishay Mansour, and Rocco A. Servedio. Agnostically learning halfspaces. *SIAM J. Comput.*, 37(6):1777–1805, 2008.

31 Michael J. Kearns and Dana Ron. Testing problems with sublearning sample complexity. *J. Comput. Syst. Sci.*, 61(3):428–456, 2000.

32 Michael J. Kearns and Leslie G. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. *J. ACM*, 41(1):67–95, 1994.

33 Subhash Khot, Dor Minzer, and Muli Safra. On monotonicity testing and Boolean isoperimetric type theorems. In *FOCS*, pages 52–58. IEEE Computer Society, 2015.

34 Pravesh Kothari, Amir Nayyeri, Ryan O'Donnell, and Chenggang Wu. Testing surface area. In *SODA*, pages 1204–1214. SIAM, 2014.

35 Chengyu Lin and Shengyu Zhang. Sensitivity conjecture and log-rank conjecture for functions with small alternating numbers. *CoRR*, abs/1602.06627, 2016.

36 A. A. Markov. On the inversion complexity of systems of functions. *Doklady Akademii Nauk SSSR*, 116:917–919, 1957. English translation in [37].

37 A. A. Markov. On the inversion complexity of a system of functions. *Journal of the ACM*, 5(4):331–334, October 1958.

38 Joe Neeman. Testing surface area with arbitrary accuracy. In *STOC*, pages 393–397. ACM, 2014.

39 Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

40 Ryan O'Donnell and Rocco A. Servedio. Learning monotone decision trees in polynomial time. *SIAM J. Comput.*, 37(3):827–844, 2007.

41 Ryan O'Donnell and Karl Wimmer. KKL, Kruskal–Katona, and monotone nets. In *FOCS*, pages 725–734. IEEE Computer Society, 2009.

42 Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Sciences*, 72(6):1012–1042, 2006.

43 Ran Raz and Avi Wigderson. Monotone circuits for matching require linear depth. *J. ACM*, 39(3):736–744, 1992.

44 Alexander A Razborov. Lower bounds on the monotone complexity of some Boolean functions. *Doklady Akademii Nauk SSSR*, 281(4):798–801, 1985.

45 Benjamin Rossman. Correlation bounds against monotone NC^1. In *Conference on Computational Complexity* (CCC), 2015.

46 Rocco A. Servedio. On learning monotone DNF under product distributions. *Inf. Comput.*, 193(1):57–74, 2004.

47 List of open problems in sublinear algorithms: Problem 70, 2016. Originally posed in [7]. URL: `http://sublinear.info/70`.

48 Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

What Circuit Classes Can Be Learned with Non-Trivial Savings?

Rocco A. Servedio[*1] **and Li-Yang Tan**[†2]

1 **Department of Computer Science, Columbia University, New York, USA**
rocco@cs.columbia.edu
2 **Toyota Technological Institute, Chicago, USA**
liyang@cs.columbia.edu

Abstract

Despite decades of intensive research, efficient – or even sub-exponential time – distribution-free PAC learning algorithms are not known for many important Boolean function classes. In this work we suggest a new perspective on these learning problems, inspired by a surge of recent research in complexity theory, in which the goal is to determine whether and how much of a savings over a naive 2^n runtime can be achieved.

We establish a range of exploratory results towards this end. In more detail,

1. We first observe that a simple approach building on known uniform-distribution learning results gives non-trivial distribution-free learning algorithms for several well-studied classes including AC^0, arbitrary functions of a few linear threshold functions (LTFs), and AC^0 augmented with mod_p gates.
2. Next we present an approach, based on the method of random restrictions from circuit complexity, which can be used to obtain several distribution-free learning algorithms that do not appear to be achievable by approach (1) above. The results achieved in this way include learning algorithms with non-trivial savings for LTF-of-AC^0 circuits and improved savings for learning parity-of-AC^0 circuits.
3. Finally, our third contribution is a generic technique for converting lower bounds proved using Nečiporuk's method to learning algorithms with non-trivial savings. This technique, which is the most involved of our three approaches, yields distribution-free learning algorithms for a range of classes where previously even non-trivial uniform-distribution learning algorithms were not known; these classes include full-basis formulas, branching programs, span programs, etc. up to some fixed polynomial size.

1998 ACM Subject Classification I.2.6 Learning

Keywords and phrases computational learning theory, circuit complexity, non-trivial savings

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.30

1 Introduction

Simple concepts are easy to learn, while complicated ones are harder to learn. Some of the earliest and most fundamental work in computational learning theory aims at elucidating this truism from a theoretical computer science perspective: can we understand how the algorithmic complexity of learning Boolean functions (i.e. the running time required by

* The author is supported by NSF grants CCF-1420349 and CCF-1563155.
† The author is supported by NSF grant CCF-1563122; this research was done while visiting Columbia University.


© Rocco A. Servedio and Li-Yang Tan;
licensed under Creative Commons License CC-BY

learning algorithms) scales with the computational complexity of the functions being learned? Achieving such an understanding was first articulated as an explicit goal in (indeed, arguably *the* explicit goal of) Valiant's landmark paper "A theory of the learnable" [42]:

> "The results of learnability theory would then indicate the maximum granularity of the single concepts that can be acquired without programming. In summary, this paper attempts to explore the limits of what is learnable as allowed by algorithmic complexity. The identification of these limits is a major goal of the line of work proposed in this paper."

So, more than thirty years later, how has learning theory fared in achieving these goals? Perhaps disappointingly, the roster of concept classes (classes of Boolean functions over $\{0,1\}^n$) for which efficient learning algorithms have been developed in Valiant's original distribution-independent PAC learning model, or in other similarly general learning frameworks, is quite short. Classes that are known to be learnable in polynomial time include linear threshold functions (LTFs) and degree-k polynomial threshold functions for $k = O(1)$ [9] (subsuming the classes of k-CNF, k-DNF [42] and k-decision lists [38]); parity functions and $\mathbb{F}_2$ polynomials of constant degree [20, 13], and not much more. (If membership queries are allowed, then a few other classes are known to be distribution-independent PAC learnable in polynomial time, such as decision trees of polynomial size [10, 6] and regular languages computed by polynomial-size DFAs [1].) In fact, only a relatively small number of natural Boolean function classes are additionally known to be learnable even if we only require sub-exponential time for the learning. DNF formulas of $\mathrm{poly}(n)$ size can be learned in $2^{\tilde{O}(n^{1/3})}$ time [27], $\mathrm{poly}(n)$-sparse $\mathbb{F}_2$ polynomials can be learned in $2^{\tilde{O}(n^{1/2})}$ time [19], and de Morgan formulas of size s can be learned in time $n^{O(s^{1/2})}$ [37].

Even simple generalizations of the above-mentioned subexponential-time-learnable function classes have remained frustratingly out of reach for the distribution-independent PAC model. Prominent examples here include the class of $\mathrm{poly}(n)$-size depth-3 AC^0 circuits, and intersections of even just two LTFs over $\{0,1\}^n$: despite extensive research effort, no positive algorithmic results are known for these classes (hence, needless to say, for their generalizations as well). This is quite a disappointment, given the rich variety of natural Boolean function classes that have been intensively studied in concrete complexity over the past several decades: well-known examples include AC^0 (augmented in various ways with more exotic gates such as mod_p gates, majority gates, threshold gates, and the like), SYM^+ circuits, various classes of branching programs, functions of a few LTFs, and more. These functions play a starring role in concrete complexity theory, but learning theorists cannot even score an autograph.[1]

This work: A change of perspective

In this paper we propose a new point of view on the challenging learning problems discussed above. Since the quest for polynomial or even sub-exponential time distribution-independent learning algorithms has been unsuccessful, we suggest that a more fruitful perspective may be to *study the question of whether, and how much of, a savings over a naive* 2^n *running time can be achieved for these learning problems.* We thus are interested in obtaining learning

[1] We note that in the (significantly) easier *uniform-distribution* learning model, in which the learner need only succeed w.r.t. the uniform distribution over $\{0,1\}^n$, positive learning results are known for functions of a few LTFs [26, 16] and AC^0 [29] as well as some generalizations of these classes [15, 11]; we will have occasion to revisit these results later.

algorithms that run in time $2^{n-s(n)}$ for some *savings* function $s(n)$ which is as large as possible. (To use a hackneyed metaphor, instead of expecting a brim-full glass, we are now hoping for a mouthful of water at the bottom...)

While this appears to be a new lens through which to view learning problems, we stress that the point of view which we advocate here has been a mainstay in computational complexity for a long time. Well-known results give non-trivial upper bounds (mildly better than 2^n) on the running time of satisfiability or (exact) counting algorithms for k-CNFs, general CNFs, and a host of NP-hard or #P-hard problems (see e.g. [33, 39, 32, 40, 14, 21, 4, 7, 22]). Williams's breakthrough connection [43, 44] linking non-trivial savings of satisfiability algorithms to circuit lower bounds has intensified the interest in results of this sort for richer circuit classes, and even more recently there has been a surge of interest in questions of a similar flavor because of the connections that have been established between hypotheses like SETH and prominent questions in algorithm design (see e.g. the survey of [45]).

In this paper we explore some first questions in the study of what can be learned with non-trivial savings; happily, it turns out that this new perspective yields a rich bounty of positive results. As our main contribution, we present three techniques and show how each technique yields new learning algorithms of the sort we are interested in. Cumulatively, our results achieve the first non-trivial savings for many well-studied circuit classes; however, several natural questions about learning with non-trivial savings are left open by our work. We hope (and expect) that further results extending our knowledge of "non-trivially learnable" function classes will follow.

A quick and dirty proof of concept

Before describing our main results, for the sake of intuition we sketch a simple argument showing that AC^0 indeed admits a non-trivial distribution-free learning algorithm (more precisely, one whose running time is $2^{n-n^{\Omega(1/d)}}$ for poly(n)-size depth-d AC^0). The argument is based on Håstad's switching lemma [17] which, roughly speaking, states that any depth-d, poly(n)-size AC^0 circuit F collapses to a shallow decision tree with very high probability under a random restriction. This can be shown to imply that if $\{0,1\}^n$ is partitioned into translations of a random subcube (corresponding to all possible settings of the live variables of a random restriction), then with very high probability almost every such subcube has the property that if F is restricted to the subcube, then F collapses to a shallow decision tree. Since it is possible to learn such a shallow decision tree relatively efficiently (in time much less than the number of points in its domain, i.e. in the subcube), this means that by learning F separately on each subcube it is possible to achieve a significant savings over brute-force search on every "good" subcube, i.e. on almost every subcube. Trading off the fraction of bad subcubes (which corresponds to the failure probability of the switching lemma, and decreases with the dimension of the subcubes) against the number of subcubes (which provides a lower bound on the running time of this learning approach, and which increases as the dimension of the subcubes decreases) and working out the parameters, the running time of this simple-minded approach comes out to be $2^{n-n^{\Omega(1/d)}}$.

Two comments: First, we note that we will improve significantly on this running time in Section 3, using a more sophisticated instantiation of this idea, and will achieve this improved running time even for various augmentations of AC^0 circuits. Second, it may not be completely clear how to run a separate copy of a distribution-free learning algorithm on each subcube in the above sketch. This will become clear in Section 2 when we describe the formal model (based on online learning, or equivalently the model of exact learning from equivalence queries) that we will use for all of our positive results (and which is well-known to imply distribution-free PAC learnability).

Relation to previous work: compression of Boolean functions

We have already explained how our goal of achieving non-trivial savings for learning is directly inspired by work aiming towards this goal for the algorithmic problems of satisfiability and counting. Another line of research which is more closely related to our study of non-trivial learning is the recent work on "compression" of Boolean functions that was initiated by [12]. A compression algorithm for a class $\mathcal{C}$ (such as the class of AC^0 circuits) is a deterministic algorithm which is given as input the 2^n-bit truth table of a function in $\mathcal{C}$, must run in time polynomial in its input length (i.e. in $2^{O(n)}$ time), and must output any Boolean circuit C, computing f, such that the size of C is less than the trivial $2^n/n$ bound.

Deterministic learning is easily seen to be at least as hard as compression; we discuss the exact relation between the two tasks in more detail in Section 2, after we have given a precise definition of our learning model. Our learning algorithms, which are randomized, imply randomized variants of almost all of the compression results in [12], in several cases with new and simpler proofs. We also establish non-trivial learning results (and hence randomized compression results) for many classes for which compression results were not previously known. These classes include LTF-of-AC^0, arbitrary functions of $o(n/\log n)$ LTFs, $n^{1.99}$-size switching networks, $n^{1.49}$-size switching-and-rectifier networks, $n^{1.49}$-size non-deterministic branching programs, and $n^{1.49}$-size span programs; in fact, for the last four of these classes we obtain deterministic compression algorithms.

1.1 Our techniques and results

To begin, in Section 2.1 we make the simple observation that uniform-distribution PAC learning algorithms can be converted to exact learning algorithms with membership queries simply by "patching up" the $\varepsilon \cdot 2^n$ points where an ε-accurate hypothesis is in error. (This observation was already employed by [11] in the context of compression.) Using known uniform-distribution learning results, this straightforward approach gives non-trivial distribution-free learning algorithms for several well-studied classes including AC^0, arbitrary functions of a few LTFs, and AC^0 augmented with mod_p gates.

However, as we explain in Section 3, there are uniform-distribution learning algorithms (such as the algorithms of [15, 24] for LTF-of-AC^0 circuits) which for technical reasons do not yield exact learning algorithms with non-trivial savings. To address this, in Section 3 we show how the method of random restrictions from circuit complexity can be employed to obtain non-trivial learning algorithms in settings where the approach of Section 2.1 does not apply. Recall that, roughly speaking, the "method of random restrictions" refers to a body of results showing that certain types of Boolean functions "collapse" to simpler functions with high probability when they are hit with a random restriction fixing a random subset of input variables to randomly chosen constant values. Our approach is based on learning the simpler functions that result from random restriction and thereby obtaining an overall savings in learning the original unrestricted function. This is similar to the "quick and dirty" proof of concept sketched earlier, but by adapting a recent powerful "multi-switching" lemma of Håstad [18] to our learning context, we are able to achieve a significantly better savings than the "quick and dirty" argument which uses only the original [17] switching lemma. Via this approach we obtain exact learning algorithms for LTF-of-AC^0 and parity-of-AC^0 that match the savings of our learning algorithm for AC^0 from Section 2.1. As indicated above the uniform-distribution approach of Section 2.1 does not give a result for LTF-of-AC^0, while for parity-of-AC^0 circuits our random restriction approach yields significantly improved savings over the results achieved for this class in Section 2.1. Furthermore, for both these classes

our learning algorithms based on random restrictions do not require membership queries (in contrast to the uniform-distribution based approach, which does require membership queries).

Our third and most involved technique for non-trivial learning is based on Nečiporuk's celebrated lower bound method: in Section 4 we give a generic translation of lower bounds proved using Nečiporuk's method to non-trivial exact learning algorithms. Roughly speaking, Nečiporuk's lower bounds are established by showing that low complexity functions have few subfunctions (and exhibiting explicit functions that have many subfunctions, and hence must have high complexity). We give an exact learning algorithm that achieves non-trivial savings for classes of functions that have few subfunctions. A key technical component of our learning algorithm is a pre-processing-based technique for executing many copies of the classical halving algorithm in a highly efficient amortized manner. While simple, this technique appears to be new and may be of use elsewhere. We thus obtain a single unified learning algorithm that achieves non-trivial savings for a broad range of function classes, including full-basis binary formulas of size $n^{1.99}$, branching programs of size $n^{1.99}$, switching networks of size $n^{1.99}$, switching-and-rectifier network of size $n^{1.49}$, non-deterministic branching programs of size $n^{1.49}$, and span programs of size $n^{1.49}$. Our learning results recapture the [12] compression results for $n^{1.99}$-size formulas and branching programs with a new and simpler proof, and give the first compression results for the other classes of switching networks, switching-and-rectifier networks, non-deterministic branching programs, and span programs listed above.

2 Preliminaries

The learning model we consider

The distribution-independent PAC model has several parameters (a confidence parameter which is usually denoted δ, and an accuracy parameter usually denoted ε) which make precise statements of running times somewhat unwieldy. In this paper we will work in a elegant model of online mistake-bound learning [30] which is well known to be equivalent to the model of exact learning from equivalence queries only [2] and to be even more demanding than the distribution-independent PAC learning model [2, 8]. A brief description of this model is as follows: Let $\mathcal{C}$ be a class of functions from $\{0,1\}^n$ to $\{0,1\}$ that is to be learned and let $f \in \mathcal{C}$ be an unknown target function. The learning algorithm always maintains a hypothesis function $h : \{0,1\}^n \to \{0,1\}$ (more precisely, a representation of h in the form of a Boolean circuit computing h). The learning process unfolds in a sequence of *trials*: at the start of a given trial,

- If $h(x) = f(x)$ for all $x \in \{0,1\}^n$ then the learning algorithm has succeeded and the process stops.
- Otherwise a *counterexample* – an arbitrary x such that $h(x) \neq f(x)$ – is presented to the learning algorithm, and the learning algorithm may update its hypothesis h before the start of the next trial.

The running time of a learning algorithm in this framework is simply the worst-case running time until the algorithm succeeds (taken over all $f \in \mathcal{C}$ and all possible sequences of counterexamples). We will also sometimes have occasion to consider an extension of this model in which at each trial the learning algorithm may, instead of receiving a counterexample, at its discretion choose instead to make a *membership query* (i.e. to submit a string $x \in \{0,1\}^n$ of its choosing to the oracle, and receive the value $f(x)$ in response). This is the well-studied framework of "exact learning from membership and equivalence queries" [2].

We will also have occasion to consider randomized exact learning algorithms. We say that a randomized algorithm learns class $\mathcal{C}$ in time $T(n)$ if for any target function $f \in \mathcal{C}$, the algorithm succeeds with probability at least $1 - \delta$ (over its internal coin tosses) after at most $T(n) \cdot \log(1/\delta)$ time steps. While many of the learning results we present will be for randomized exact learning algorithms, in the rest of this section for simplicity we confine our discussion to deterministic learning algorithms.

Besides being a clean and attractive learning model, learnability in the exact learning model (optionally augmented with membership queries) is well known to imply learnability in the distribution-independent PAC model (correspondingly augmented with membership queries). More precisely, if a class $\mathcal{C}$ is learnable in time $T(n)$ using $Q(n)$ queries in the exact model, then by a standard argument[2] $\mathcal{C}$ is learnable to confidence $1 - \delta$ and accuracy $1 - \varepsilon$ in the PAC model in time $T_{\mathrm{PAC}} = O(\frac{T(n)}{\varepsilon} \ln(\frac{T(n)}{\delta}))$ using $O(\frac{Q(n)}{\varepsilon} \ln(\frac{Q(n)}{\delta}))$ queries.

Non-trivial savings

It is easy to see that any class $\mathcal{C}$ can be learned in time $\mathrm{poly}(n) \cdot 2^n$ in our model via a simple memorization-based approach; our goal in this work will be to come up with algorithms whose running time is $2^{n-s(n)}$ where the *savings* $s(n)$ is as large as possible. We say that any savings function $s(n) = \omega(\log n)$ is *non-trivial*. We observe that the conversion from exact learning to distribution-independent PAC learning described above preserves learnability with non-trivial savings: learnability with non-trivial savings in our exact model implies learnability to any $1/\mathrm{poly}(n)$ accuracy and confidence in the PAC model with non-trivial savings (since if $T(n) = 2^{n-\omega(\log n)}$ and $\varepsilon, \delta = 1/\mathrm{poly}(n)$ then $T_{\mathrm{PAC}} = 2^{n-\omega(\log n)}$).

Deterministic learning implies compression

As mentioned in the introduction, now that we have a precise definition of our learning model it is easy to verify that any class of functions that admits a non-trivial deterministic learning algorithm admits a compression algorithm. To see this, observe that our learning algorithms (i) are not given the full truth table of f as input, and (ii) must run in time strictly less than 2^n (as opposed to $2^{O(n)}$ for compression), while (iii) a learning algorithm in our framework must (like a compression algorithm) ultimately construct a circuit computing f that has size less than $2^n/n$. We may summarize this discussion in the following observation:

▶ **Observation 1.** *Let $\mathcal{C}$ be a class of n-variable Boolean functions that has a deterministic exact learning algorithm using membership and equivalence queries with savings $s(n) = \omega(\log n)$ (i.e. in time $2^{n-s(n)}$). Then there is a deterministic algorithm that compresses $\mathcal{C}$ to circuits of size $2^{n-s(n)}$.*

As an application of this observation, consider the class of size-S read-once branching programs (ROBPs) over $x_1, \ldots, x_n$. Since every such size-S ROBP is a deterministic finite automaton with S nodes over the binary alphabet $\{0, 1\}$ accepting only n-bit strings, Angluin's deterministic exact learning algorithm [1] (which uses membership and equivalence queries) can learn any such ROBP in time $O(S^2)$. By Observation 1, this implies an algorithm that compresses $2^{n/2-s(n)/2}$-size ROBPs to circuits of size $O(2^{n-s(n)})$. This recovers a compression result for this class that was previously obtained by Chen et al. in the paper [12] that initiated the study of compression algorithms (see their Theorem 3.8).

---

[2] See e.g. Section 2.4 of [2], replacing the expression "$i \ln 2$" by "$\ln(2i^2)$".

2.1 A first simple approach based on uniform-distribution learning

In contrast with the state of affairs for distribution-independent PAC learning, a more significant body of results is known for *uniform-distribution* PAC learning (as we will see later in this section). In this section we describe a simple approach by which some uniform-distribution PAC learning algorithms – roughly speaking, those which have a good dependence on the accuracy parameter ε – can easily be translated into non-trivial exact learning algorithms.

The simple approach, which was already suggested in [11] in the context of compression, is as follows. Using membership queries, we may simulate uniform random examples $(\boldsymbol{x}, f(\boldsymbol{x}))$ and run the uniform-distribution learning algorithm to obtain an ε-accurate hypothesis h in time $T(n, 1/\varepsilon, \log(1/\delta))$. Then we use at most $\varepsilon \cdot 2^n$ equivalence queries to identify and correct all of the (at most $\varepsilon \cdot 2^n$) many points on which h is incorrect. Since updating the hypothesis after each equivalence query can clearly be done in time $\mathrm{poly}(n)$ we thus obtain the following:

▶ **Claim 2.** *Let $\mathcal{C}$ be a class of n-variable Boolean circuits such that there is a uniform-distribution PAC learning algorithm (which may possibly use membership queries) running in time $T(n, 1/\varepsilon, \log(1/\delta))$, which with probability $1-\delta$ outputs an ε-accurate hypothesis. Then there is a randomized exact learning algorithm for $\mathcal{C}$ which uses membership and equivalence queries and runs in time*

$$\mathrm{poly}(n) \cdot \min_{\varepsilon > 0} \left\{ T(n, 1/\varepsilon, \log(1/\delta)) + \varepsilon \cdot 2^n \right\}. \tag{1}$$

First application of Claim 2: Learning AC^0 circuits

The seminal work of Linial, Mansour, and Nisan [29] established Fourier concentration bounds for size-M depth-d circuits, and showed how these bounds straightforwardly yield uniform-distribution learning algorithms. An essentially optimal strengthening of the Fourier concentration bound of [29] was recently obtained by Tal [41], who showed that there exists a universal constant $c > 0$ such that every size-M depth-d circuit C satisfies $\sum_{|S| \geq c \log^{d-1}(M) \log(1/\varepsilon)} \widehat{C}(S)^2 \leq \varepsilon$. Via the connection between Fourier concentration and uniform-distribution learning established by [29], this implies that the class of size-M depth-d circuits can be learned to accuracy ε by a randomized algorithm in time $\mathrm{poly}\left(\binom{n}{\leq c \log^{d-1}(M) \log(1/\varepsilon)}\right) \cdot \log(1/\delta)$. Consequently, by Claim 2, taking $\varepsilon = 2^{-\Theta(n/(\log M)^{d-1})}$ we get a randomized exact learning algorithm which uses membership and equivalence queries which runs in time $\mathrm{poly}(n) \cdot 2^{n - \Omega(n/(\log M)^{d-1})}$. We note that this matches the circuit size given by the compression theorem of [12] for such circuits.

Learning functions of k LTFs

Gopalan et al. [16] have given a randomized uniform-distribution membership-query algorithm that learns any function of k LTFs over $\{0,1\}^n$ in time $O((nk/\varepsilon)^{k+1})$. Choosing $\varepsilon = 2^{-\frac{n}{k+2}}$ in Claim 2, we get a randomized exact learning algorithm which uses membership and equivalence queries and runs in time $\mathrm{poly}(n) \cdot 2^{\frac{k+1}{k+2} n} = \mathrm{poly}(n) \cdot 2^{n - \frac{n}{k+2}}$, thus achieving a non-trivial savings for any $k = o(n/\log n)$, and a linear savings for any constant k.

Learning $\mathsf{AC}^0[p]$ circuits

A recent exciting result of [11] gives a randomized uniform-distribution membership-query algorithm for learning the class of n-variable size-M depth-d $\mathsf{AC}^0[p]$ circuits to accuracy ε

in time $2^{(\log(Mn/\varepsilon))^{O(d)}}$. By Claim 2, taking $\varepsilon = M \cdot 2^{-n^{\Theta(1/d)}}$, we get a randomized exact learning algorithm which uses membership and equivalence queries and runs in time $2^{n-n^{\Omega(1/d)}}$ for all circuits of size $M \leq 2^{n^{c/d}}$ for some absolute constant $c > 0$.

3 Beyond uniform-distribution learnability: Learning via random restrictions

As noted briefly in Section 2.1, in order for Claim 2 to give a non-trivial savings for exact learning the running time $T(n, 1/\varepsilon, \log(1/\delta))$ of the uniform-distribution learning algorithm must not depend too badly on $1/\varepsilon$. This requirement limits the applicability of Claim 2; to see a concrete example of this, let us consider the class of all poly(n)-size, depth $d = O(1)$ LTF-of-AC^0 circuits (so such a circuit has an arbitrary linear threshold function as the output gate with poly(n) many poly(n)-size depth-$(d-1)$ AC^0 circuits feeding into the threshold gate). Uniform-distribution learning results [15, 24] are known for this class, based on Fourier concentration which is established via known upper bounds on the average sensitivity of low-degree polynomial threshold functions. As discussed in [15], the best running time that can be achieved for learning via this approach is $n^{(\log n)^{O(d)}/\varepsilon^2}$, which would follow from a conjecture of Gotsman and Linial, known to be best possible, upper bounding the average sensitivity of low-degree polynomial threshold functions. (The current state of the art learning algorithms, based on Kane's upper bound [24] on the average sensitivity of low-degree polynomial threshold functions which nearly matches the Gotsman-Linial conjecture, have a slightly worse running time.) As a result of this poor dependence on ε, the value of (1) is $\Omega(2^n/\sqrt{n})$, so no non-trivial savings is achieved. We note that even for the $d = 1$ case of a single linear threshold gate as the function to be learned, the best possible running time of a learning algorithm based on Fourier concentration is $n^{\Omega(1/\varepsilon^2)}$ (see Theorem 23 of [26]).

An approach based on random restrictions

In this section we show that by taking a more direct approach than Claim 2, it is possible to achieve a non-trivial savings for LTF-of-AC^0 circuits, and to improve on the results achievable via Claim 2 for the class of Parity-of-AC^0 circuits, which is covered by the final learning result in Section 2.1. An additional advantage of this random restriction based approach is that (unlike the uniform distribution approach based on Claim 2) the resulting exact learning algorithms do not require membership queries, only equivalence queries.

This approach is based on the method of random restrictions; it is reminiscent of the simple "proof of concept" from the Introduction (though we will ultimately instantiate it with a more sophisticated switching lemma than Håstad's original switching lemma [17]). Roughly speaking the approach works as follows: Let $\mathcal{R}_p$ denote the distribution over n-variable random restrictions (i.e. over $\{0, 1, *\}^n$) that independently sets each coordinate to 0, 1, or $*$ with probabilities $\frac{1-p}{2}$, $\frac{1-p}{2}$ and p respectively. Let $\mathcal{C}$ be the class of functions that we would like to learn, and let $\mathcal{C}'$ be some other class of functions (which should be thought of as "simpler" than the functions in $\mathcal{C}$). If we have (i) a switching lemma type statement establishing that for $\boldsymbol{\rho} \leftarrow \mathcal{R}$, any $f \in \mathcal{C}$ with high probability collapses under $\boldsymbol{\rho}$ to a function in $\mathcal{C}'$, and (ii) an exact algorithm A that can learn functions in $\mathcal{C}'$ in time significantly faster than brute force, then we can achieve nontrivial savings by (a) drawing a random restriction $\boldsymbol{\rho} \leftarrow \mathcal{R}$, (b) partitioning $\{0,1\}^n$ into translates of the $|\boldsymbol{\rho}^{-1}(*)|$-dimensional subcube corresponding to the unfixed variables of $\boldsymbol{\rho}$, and (c) running the algorithm A on each of the $2^{n-|\boldsymbol{\rho}^{-1}(*)|}$ many such subcubes. By (i), for most subcubes we will achieve a significant savings over a brute-force $2^{|\boldsymbol{\rho}^{-1}(*)|}$ running time for that subcube; even "paying

full fare" for the (few) remaining bad subcubes, this results in an overall algorithm with non-trivial savings.

We make this discussion formal in the following lemma:

▶ **Lemma 3.** *Let $\mathcal{C}$ and $\mathcal{C}'$ be two classes of Boolean functions, where $\mathcal{C} = \bigcup_{n\geq 1} \mathcal{C}_n$ and functions in $\mathcal{C}_n$ are n-variable Boolean functions and likewise for $\mathcal{C}'$. Suppose that $\mathcal{C}$ and $\mathcal{C}'$ are such that for some value $\frac{8}{n} \leq p < 1$, we have*

1. *(switching lemma from $\mathcal{C}$ to $\mathcal{C}'$) For every function $f \in \mathcal{C}_n$,*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{\rho}\leftarrow\mathcal{R}_p}\left[f \restriction \boldsymbol{\rho} \text{ does not belong to } \mathcal{C}'_{|\boldsymbol{\rho}^{-1}(*)|}\right] \leq \alpha(n); \tag{2}$$

2. *(efficient learnability of $\mathcal{C}'$) There is an exact learning algorithm A for $\mathcal{C}'$ that uses equivalence queries only and runs in time $T(\ell) = 2^{o(\ell)}$ when it is run on a function in $\mathcal{C}'_\ell$.*

Then there is a randomized exact learning algorithm for $\mathcal{C}_n$ which uses equivalence queries only, outputs a correct hypothesis with probability $1-\delta$, and runs in time

$$\mathrm{poly}(n) \cdot \left(2^{n-pn/2} \cdot T(pn/2) + \alpha(n) \cdot 2^n\right) \cdot \log(1/\delta). \tag{3}$$

Proof. The randomized exact learning algorithm executes a sequence of at most $O(\log(1/\delta))$ independent stages, halting the first time a stage succeeds. We will show below that each stage succeeds in producing an exactly correct hypothesis with probability at least 0.35, and runs in time $\mathrm{poly}(n) \cdot \left(2^{n-pn/2} \cdot T(pn/2) + \alpha(n) \cdot 2^n\right)$; the lemma follows easily from this.

Each stage consists of two substages and is structured as follows. In the first substage, the exact learning algorithm draws a random restriction $\boldsymbol{\rho} \leftarrow \mathcal{R}_p$. By a standard multiplicative Chernoff bound (using $p \geq \frac{8}{n}$) we have that $|\boldsymbol{\rho}^{-1}(*)| < pn/2$ with probability at most $\exp(-pn/8) < e^{-1}$; if $|\boldsymbol{\rho}^{-1}(*)| < pn/2$ then this stage ends in failure, otherwise the algorithm continues to the second substage (described in the next paragraph). Let $C_{\boldsymbol{\rho}}$ be the subcube of $\{0,1\}^n$ (of dimension $|\boldsymbol{\rho}^{-1}(*)|$ and containing $2^{|\boldsymbol{\rho}^{-1}(*)|}$ many points) corresponding to the live variables of $\boldsymbol{\rho}$, and let $C_{\boldsymbol{\rho},\text{translates}}$ be the set of all $2^{|n-\boldsymbol{\rho}^{-1}(*)|}$ many disjoint translates of $C_{\boldsymbol{\rho}}$ which together cover $\{0,1\}^n$. We say that a translate $C_{\boldsymbol{\rho}} + z \in C_{\boldsymbol{\rho},\text{translates}}$ (viewing addition as being over $\mathbb{F}_2$) of $C_{\boldsymbol{\rho}}$ is *bad* if the translated restriction $\boldsymbol{\rho} + z$ (whose $*$'s are in the exact same locations as those of $\boldsymbol{\rho}$) corresponding to $C_{\boldsymbol{\rho}} + z$ is such that $f \restriction (\boldsymbol{\rho} + z)$ does not belong to $\mathcal{C}'$, and we say that $\boldsymbol{\rho}$ is bad if more than a $4\alpha(n)$ fraction of the $2^{|n-\boldsymbol{\rho}^{-1}(*)|}$ translates of $C_{\boldsymbol{\rho}}$ are bad. By Markov's inequality applied to (2), we have that $\boldsymbol{\rho} \leftarrow \mathcal{R}_p$ is bad with probability at most $1/4$. We thus have that with overall probability at least $1 - 1/4 - e^{-1} > 0.35$ over the draw of $\boldsymbol{\rho} \leftarrow \mathcal{R}_p$, the stage proceeds to the second substage with a restriction $\boldsymbol{\rho}$ that is not bad (and that satisfies $|\boldsymbol{\rho}^{-1}(*)| \geq pn/2$).

In the second substage, the exact learning algorithm then runs $2^{|n-\boldsymbol{\rho}^{-1}(*)|}$ copies of algorithm A in parallel, each one to learn the $(\ell = |\boldsymbol{\rho}^{-1}(*)|)$-variable function which is $f \restriction (C_{\boldsymbol{\rho}} + z)$ for one of the translates of $C_{\boldsymbol{\rho}}$. This can be done using equivalence queries only: the overall hypothesis at each time step is obtained from the $2^{|n-\boldsymbol{\rho}^{-1}(*)|}$ many hypotheses (one for each subcube) in the obvious way. Each counterexample received allows one of the $2^{|n-\boldsymbol{\rho}^{-1}(*)|}$ copies of algorithm A (the one running over the subcube that received the counterexample) to update its hypothesis. Let $M(\ell) \leq T(\ell)$ be the maximum number of counterexamples that A can ever receive when it is run on a function in $\mathcal{C}'_\ell$. Within each subcube, if the copy of A running in that subcube receives more than $M(\ell)$ counterexamples, then since that subcube must be bad, the overall exact learning algorithm switches from running A on that subcube to running a naive equivalence-query learning algorithm that simply builds a truth table (and takes time at most $\mathrm{poly}(n) \cdot 2^{|\boldsymbol{\rho}^{-1}(*)|}$, the number of points in the subcube).

The second substage carries out this process until either

(i) no counterexample is provided (meaning that all $2^{|n-\boldsymbol{\rho}^{-1}(*)|}$ copies of the algorithm have obtained an exactly correct hypothesis, and thus the overall combined hypothesis is exactly correct and the stage succeeds), or

(ii) more than $4\alpha(n)2^{n-\ell}$ copies of the algorithm have each received more than $M(\ell)$ counterexamples; since this can only happen if $\boldsymbol{\rho}$ is bad, in this case the stage halts and ends in failure.

We observe that case (i) must occur if $\boldsymbol{\rho}$ is not bad, and hence case (i) occurs and the stage succeeds with overall probability at least 0.35. In either case the running time for the stage is at most

$$\begin{aligned}
&\mathrm{poly}(n) \cdot \left(2^{|n-\ell|} \cdot T(\ell) + 4\alpha(n)2^{n-\ell} \cdot (T(\ell) + 2^{\ell})\right) \\
&< \mathrm{poly}(n) \cdot \left(2^{|n-\ell|} \cdot T(\ell) + 4\alpha(n)2^{n-\ell} \cdot (2 \cdot 2^{\ell})\right) \\
&< \mathrm{poly}(n) \cdot \left(2^{n-pn/2} \cdot T(pn/2) + \alpha(n) \cdot 2^{n}\right)
\end{aligned}$$

time steps, where the first summand on the LHS upper bounds the total running time of all the learning algorithms that are running over non-bad subcubes, and the second summand bounds the total running time of all the learning algorithms that are running over the (at most $4\alpha(n)2^{n-\ell}$) many bad subcubes. As discussed at the beginning of the proof, this establishes the lemma. ◀

3.1 An application of Lemma 3: learning LTF-of-AC^0 and Parity-of-AC^0

In this subsection we use Lemma 3 to obtain non-trivial exact learning algorithms for LTF-of-AC^0 and Parity-of-AC^0 circuits. As discussed at the start of Section 3, it does not seem possible to obtain a non-trivial exact learning algorithm for LTF-of-AC^0 using known uniform-distribution learning results. The learning algorithm for Parity-of-AC^0 that we give in this subsection achieves significantly better savings than the algorithm from Section 2.1, and moreover does not require membership queries.

In order to apply Lemma 3 we need a suitable switching lemma from $\mathcal{C}$ to $\mathcal{C}'$ and a learning algorithm for $\mathcal{C}'$. Looking ahead, for LTF-of-AC^0 the class $\mathcal{C}'$ will be the class of low-degree polynomial threshold functions, and for Parity-of-AC^0 it will be the class of low-degree $\mathbb{F}_2$ polynomials. We can use the same switching lemma for both results; to describe the switching lemma we need, we recall some terminology from [18]. Let $\mathcal{G}$ be a family of Boolean functions. A decision tree T is said to be a *common ℓ-partial decision tree for $\mathcal{G}$* if every $g \in \mathcal{G}$ can be expressed as T with depth-ℓ decision trees hanging off its leaves. (Equivalently, for every $g \in \mathcal{G}$ and root-to-leaf path π in T, we have that $g \restriction \pi$ is computed by a depth-ℓ decision tree.)

If g is a Boolean function and $\mathcal{C}$ is a class of circuits, we say that g is *computed by a $(t, \mathcal{C})$-decision tree* if g is computed by a decision tree of depth t (with single Boolean variables x_i at internal nodes as usual) in which each leaf is labeled by a function from $\mathcal{C}$. We write DT_k to denote the class of depth-k decision trees.

We use a recent powerful switching lemma for multiple DNFs due to Håstad [18] (a similar switching lemma was independently obtained by [21]):[3]

[3] We note that this multi-switching lemma is the key technical ingredient in [41]'s sharpening of the [29] Fourier concentration result which gave our AC^0 learning result in Section 2.1.

▶ **Theorem 4** ([18] multi-switching lemma). *Let $\mathcal{F} = \{F_1, \ldots, F_S\}$ be a collection of depth-2 circuits with bottom fan-in w. Then for any $t \geq 1$,*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{\rho} \leftarrow \mathcal{R}_p} \left[\mathcal{F} \restriction \boldsymbol{\rho} \text{ does not have a common } (\log S)\text{-partial DT of depth} \leq t \right] \leq S(24pw)^t.$$

We will use the following simple corollary for AC^0 circuits augmented with some gate G on top as our "switching lemma from $\mathcal{C}$ to $\mathcal{C}'$" in Lemma 3 (see Appendix A for the proof):

▶ **Corollary 5.** *Let G be any Boolean function, and let F be a size-S depth-$(d+1)$ $G \circ \mathsf{AC}^0$ circuit (where we view G as a single gate at the output of the circuit). Then for $p = \frac{1}{48}(48 \log S)^{-(d-1)}$ and any $t \geq 1$,*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{\rho} \leftarrow \mathcal{R}_p} \left[F \restriction \boldsymbol{\rho} \text{ is not computed by a } (2^d t, G \circ \mathsf{DT}_{\log S})\text{-decision tree} \right] \leq d \cdot S \cdot 2^{-t}.$$

For the exact learning results we need, we recall the following well-known facts (the first follows easily from [31], see e.g. [19], and the second follows easily from Gaussian elimination):

▶ **Fact 6.**

1. *There is an exact learning algorithm (using equivalence queries only) that learns degree-d polynomial threshold functions (PTF) over ℓ Boolean variables in time* $\mathrm{poly}(\binom{\ell}{\leq d})$.
2. *The same running time holds for exact learning degree-d $\mathbb{F}_2$ polynomials (again using equivalence queries only).*

All the pieces are now place for our exact learning algorithms for LTF-of-AC^0 and parity-of-AC^0:

▶ **Theorem 7.**

1. *There is an exact learning algorithm (using equivalence queries only) that learns the class of size-S depth-$(d+1)$ LTF-of-AC^0 circuits over $\{0,1\}^n$ in time $S \cdot 2^{n - n/O(\log S)^{d-1}}$.*
2. *The same running time holds for exact learning size-S depth-$(d+1)$ Parity-of-AC^0 (again using equivalence queries only).*

Proof. We prove part (1) first (part (2) is almost identical). Let $\mathcal{C}'$ be the class of all PTFs of degree $2^d t + \log S$ (where t will be chosen later). We observe that any $\mathsf{LTF} \circ \mathsf{DT}_{\log S}$ circuit computes a PTF of degree $\log S$, and moreover that any $(2^d t, \mathsf{LTF} \circ \mathsf{DT}_{\log S})$-decision tree computes a PTF of degree $2^d t + \log S$. Applying part (1) of Fact 6, Corollary 5, and Lemma 3 with p as in Corollary 5 and choosing $t = 0.1pn/(2 \cdot 2^d)$, we get the desired learning algorithm. Part (2) follows similarly but now using the observation that any $(2^d t, \mathsf{PAR} \circ \mathsf{DT}_{\log S})$-decision tree computes an $\mathbb{F}_2$ polynomial of degree $2^d t + \log S$. ◀

4 Learning with non-trivial savings via Nečiporuk's method

In this section we present our third technique for learning with non-trivial savings. This technique is based on Nečiporuk's method, which gives a lower bound on the complexity of a function f (in various computational models such as formula size, branching program size, etc.) in terms of the number of subfunctions of f. In more detail, Nečiporuk's theorem essentially says that if the variables of f can be partitioned into disjoint subsets $S_1, S_2, \ldots$ such that the product, across all i, of (the number of distinct subfunctions than can arise when all variables in $[n] \setminus S_i$ are fixed to constants in all $2^{n-|S_i|}$ possible ways) is large, then f must have high complexity. Our technique is based on a contrapositive view: if f is a function of "not too high" complexity, then in any partition of the variables into

"large" equal-size subsets, $S_1, S_2, \ldots$ there must be some S_i over which f has "not too many" distinct subfunctions – in particular, far fewer than $2^{n-|S_i|}$, the number of distinct subcubes corresponding to the restrictions that fix all variables in $[n] \setminus S_i$. We show that this structure (having "few" subfunctions over a "large" subset of variables) can be exploited to learn f with non-trivial savings.

Warmup: Compression

In Section 4.1 we first develop this idea for the easier problem of compression rather than learning. We obtain a new and simpler algorithm and analysis recovering the deterministic compression results of [12] for $n^{1.99}$-size full-basis binary formulas and $n^{1.99}$-size branching programs. ([12] had to develop new high-probability analyses of shrinkage under random restrictions using novel martingale arguments and combine these analyses with a generalization of the greedy set-cover heuristic, whereas we only use the statement of Nečiporuk's theorem in a black-box way together with short and elementary arguments.) Thanks to the generality of Nečiporuk's method, our algorithm and analysis also yields new deterministic compression results for switching networks of size $n^{1.99}$, switching-and-rectifier networks of size $n^{1.49}$, non-deterministic branching programs of size $n^{1.49}$, and span programs of size $n^{1.49}$.

Learning

Progressing from compression to learning, next in Section 4.2 we describe how pre-processing can be used to create a data structure which enables a highly efficient implementation of the classic "halving algorithm" from learning theory. While a naive implementation of the halving algorithm to learn an unknown function from a class of N functions over an M-element domain takes time $O(NM)$, we show that by first carrying out a pre-processing step taking time $M^{O(\log N)}$ it is possible to run the halving algorithm in time only $\mathrm{poly}(\log N, \log M)$, an *exponential* savings. This means that if we need to run the halving algorithm many times, by first running the pre-processing step (which needs to be done only once) we can carry out these many runs of the halving algorithm in a highly efficient *amortized* way. Intuitively, running the halving algorithm many times is precisely what we need to do in our Nečiporuk-based learning approach: if S is the "large" subset of variables such that f has "not too many" subfunctions over S, then we will run the halving algorithm $2^{n-|S|}$ times, once for each possible subcube keeping the variables in S free, to learn the corresponding $2^{n-|S|}$ different restrictions of f.

Finally, in Section 4.3 we describe and analyze our general learning algorithm based on Nečiporuk's method. The algorithm has three stages: in the first stage, membership queries are used to randomly sample subcubes corresponding to S, which are exhaustively queried to learn the subfunctions they contain. In this way the first stage constructs a set A containing all "important" subfunctions (ones that occur in "many" subcubes); crucially, thanks to the Nečiporuk argument, the set is not too large (since there are "few" distinct subfunctions in total, important or otherwise). The second stage performs the above-described pre-processing on the set A of subfunctions, and the third stage runs the halving algorithm over all $2^{n-|S|}$ subcubes corresponding to S in the efficient amortized way described above. This results in a hypothesis which is exactly correct on every subcube containing an "important" subfunction; by definition there are only "few" subcubes that contain non-important subfunctions, and the hypothesis can be patched up on those subcubes at relatively small cost.

4.1 Compression based on having few subfunctions

Given $f : \{0,1\}^n \to \{0,1\}$ and $S \subseteq [n]$, let $\mathcal{R}_S$ denote the set of all $2^{n-|S|}$ restrictions that leave precisely the variables in S free and assign either 0 or 1 to each element of $[n] \setminus S$ in all possible ways. Let $\mathrm{Num}(f, S)$ denote the number of distinct functions from $\{0,1\}^S$ to $\{0,1\}$ that occur in $\{f \restriction \rho\}_{\rho \in \mathcal{R}_S}$ (i.e. that occur as subfunctions of f).

▶ **Lemma 8** (Compression based on few subfunctions). *Fix any partition $S_1, S_2, \ldots, S_{n^{1-\delta}}$ of $[n]$ into equal-size subsets S_i of size n^δ each, where $\delta > 0$. Let $\mathcal{C}$ be a class of n-variable functions such that for each $f \in \mathcal{C}$ there is a set S_i such that $\mathrm{Num}(f, S_i) \leq 2^{n^\beta}$, where $\beta < 1$. Then there is a compression algorithm for $\mathcal{C}$ running in time $2^{O(n)}$ with savings n^δ (i.e. given as input the truth table of any $f \in \mathcal{C}$, the algorithm outputs a circuit computing f of size $\mathrm{poly}(n) \cdot 2^{n-n^\delta}$).*

Proof. Fix $f \in \mathcal{C}$, and say that any $i \in [n^{1-\delta}]$ for which $\mathrm{Num}(f, S_i) \leq 2^{n^\beta}$ is *good*. The compression algorithm works as follows:

1. For $i = 1, 2, \ldots$ check whether i is good by building a sorted list of all the distinct subfunctions occuring in $\{f \restriction \rho\}_{\rho \in \mathcal{R}_{S_i}}$. This can be done in time $2^{O(n)}$. The hypothesis of the lemma ensures that some i is good; in the following steps for notational simplicity we suppose that $i = 1$ is good. So at this point the algorithm has a sorted list L containing at most 2^{n^β} truth tables (each being an 2^{n^δ}-bit string), and for every $\rho \in \mathcal{R}_{S_1}$ the truth table of $f \restriction \rho$ is in the list.
2. Iterate across all $\rho \in \mathcal{R}_{S_1}$ to construct a function $\Phi : \mathcal{R}_{S_1} \to [2^{n^\beta}]$ such that for each $\rho \in \mathcal{R}_{S_1}$ the value of $\Phi(\rho)$ is the index j such that the truth table of $f \restriction \rho$ is the j-th element of the list L. (Note that the description length of the function Φ is $|\mathcal{R}_{S_1}| \cdot \log(2^{n^\beta}) = \mathrm{poly}(n) \cdot 2^{n-n^\delta}$.) This can be done in time $2^{O(n)}$.
3. Finally, the compression algorithm outputs a circuit which works as follows: given input $x \in \{0,1\}^n$, let ρ_x be the element of $\mathcal{R}_{S_1}$ that is consistent with x (fixing the variables in $[n] \setminus S_1$ according to x). The circuit outputs the appropriate output bit (corresponding to the bit-string x restricted to the coordinates in S_1) from the $\Phi(\rho_x)$-th truth table of L. This circuit computes f and is of size $\mathrm{poly}(n) \cdot (|\mathcal{R}_{S_1}| + 2^{n^\delta} \cdot |L|) = \mathrm{poly}(n) \cdot (2^{n-n^\delta} + 2^{n^\delta} \cdot 2^{n^\beta}) \leq \mathrm{poly}(n) \cdot 2^{n-n^\delta}$, and this step can be done in time $2^{O(n)}$.

◀

Given Lemma 8, a direct invocation of the lower bounds provided by Nečiporuk's method for various computational models gives the following corollary, providing a wide range of deterministic compression results. We refer the reader to [23] for detailed definitions of all the computational models mentioned in Corollary 9.

▶ **Corollary 9.** *Boolean n-variable functions computable by computational model $\mathcal{A}$ of size S are compressible in time $2^{O(n)}$ to circuits of size at most 2^{n-n^ε} for a fixed $\varepsilon > 0$, where*

1. *$\mathcal{A}$ = full-basis binary formulas, $S = n^{1.99}$;*
2. *$\mathcal{A}$ = branching programs, $S = n^{1.99}$;*
3. *$\mathcal{A}$ = switching networks, $S = n^{1.99}$;*
4. *$\mathcal{A}$ = switching-and-rectifier networks, $S = n^{1.49}$;*
5. *$\mathcal{A}$ = non-deterministic branching programs, $S = n^{1.49}$;*
6. *$\mathcal{A}$ = span programs, $S = n^{1.49}$.*

Proof. We first give the argument for (1), full-basis binary formulas and $S = n^{1.99}$. We take $\delta = 0.004$ in Lemma 8, so $1 - \delta = 0.996$. Let f be any n-variable function with a full-basis

binary formula of size at most $n^{1.99}$. We recall that Nečiporuk's lower bound for full-basis formula size of f (denoted $L(f)$) implies that

$$\frac{1}{4}\sum_{i=1}^{n^{0.996}} \log(\mathrm{Num}(f, S_i)) \leq L(f) \leq n^{1.99},$$

so there is some $i \in [n^{0.996}]$ such that $\log(\mathrm{Num}(f, S_i)) \leq 4n^{0.994} < n^{0.995}$, so we have $\beta = 0.995$ and obtain the claimed compression result from Lemma 8.

The arguments for (2) and (3) follow similarly, using

$$\tau \cdot \sum_{i=1}^{n^{0.996}} \frac{\log(\mathrm{Num}(f, S_i))}{\log\log(\mathrm{Num}(f, S_i))} \leq S(f) \leq BP(f)$$

(see e.g. Theorem 15.1 of [23]) for some absolute constant τ, where $BP(f)$ denotes the branching program size of f and $S(f)$ denotes the switching network size of f.

(4) and (5) also follow similarly, recalling that Nečiporuk's method gives

$$\frac{1}{4} \cdot \sum_{i=1}^{n^{0.996}} \sqrt{\log(\mathrm{Num}(f, S_i))} \leq RS(f) \leq NBP(f)$$

(see [34]), where $RS(f)$ denotes the rectifier-and-switching network size of f and $NBP(f)$ denotes the non-deterministic branching program size of f. Finally, for (6) we recall that

$$\frac{1}{2} \cdot \sum_{i=1}^{n^{0.996}} \sqrt{\log(\mathrm{Num}(f, S_i))} \leq SPAN(f)$$

(see Theorem 1 of [25]), where $SPAN(f)$ denotes the span program size of f. ◀

4.2 More efficient implementation of the halving algorithm via pre-processing

We begin by recalling the halving algorithm [3, 2, 30] and its running time when it is executed over a domain X of M points to learn an unknown function that is promised to belong to a set $\mathcal{C}$ of at most N (known) functions, where each function in $\mathcal{C}$ may be viewed simply as a truth table of length M. (In the context of the previous subsection the domain size M corresponds to 2^{n^δ}, the number of points in each subcube, and N corresponds to 2^{n^β}, the number of subfunctions.) Recall that after a set A of labeled examples has been received, the *version space* of A is the subset of functions in $\mathcal{C}$ that are consistent with A. At each stage in the halving algorithm's execution, its current hypothesis is the majority vote over the version space of the labeled examples received thus far. Before any counterexamples have been received, initially the version space is all of $\mathcal{C}$; thus the first thing that the halving algorithm does is spend NM time to (a) read the entire bit-matrix corresponding to the current version space $\mathcal{C}$ (think of this matrix as having N columns, which are the truth tables of the functions in the class, and M rows corresponding to the points in the domain) and (b) for each row compute and record the majority vote over the elements in this row (which is the initial hypothesis). The halving algorithm gets a counterexample, and then updates its version space; since its hypothesis was the majority vote of all functions in the previous version space, at least half of the columns (the functions that are inconsistent with the counterexample) are erased, and the size of the version space goes down by at least 1/2.

To form the next hypothesis the halving algorithm spends at most $(N/2)M$ time to read the matrix corresponding to the current version space and for each row compute and record the majority vote over the surviving elements in this row. This proceeds for at most $\log N$ steps, after which the version space must be of size one, and this sole surviving function must be the unknown target function. In the i-th stage the time required is $(N/2^i)M$ so the overall runtime is $O(NM)$. (Note that if the halving algorithm were performed separately and independently Z times (corresponding in our setting to the $Z = 2^{n-|S|}$ many distinct subcubes), the overall runtime would be $ZNM > 2^n$, which is too expensive for learning with non-trivial savings.)

The following lemma shows that the halving algorithm can be implemented *exponentially* more efficiently after an initial pre-processing stage. (Crucially, the pre-processing can be done only once even if the halving algorithm will be run many times; this leads to a tremendous amortized savings.) While simple, we are not aware of previous work giving an efficient amortized implementation of the halving algorithm.

▶ **Lemma 10.** *Given a class $\mathcal{C}$ of N functions over an M-point domain X, there is a pre-processing procedure that (i) can be carried out in time $M^{O(\log N)}$ and (ii) creates a data structure* DS *such that given access to* DS*, the halving algorithm can be run to learn an unknown $f \in \mathcal{C}$ in time* $\mathrm{poly}(\log N, \log M)$*. (Consequently, given access to* DS*, the halving algorithm can be run Z times to learn a sequence $f_1, \ldots, f_Z$ of functions from $\mathcal{C}$ in total time $Z \cdot \mathrm{poly}(\log N, \log M)$.)*

Proof. We first describe the data structure DS and then establish (i) by explaining how it can be constructed in $M^{O(\log N)}$ time. We then establish (ii) by showing how DS can be used to run the halving algorithm efficiently.

The data structure DS

We say that a *size-i sample* is a set of precisely i labeled pairs $(x^1, y^1), \ldots, (x^i, y^i)$ where $x^1, \ldots, x^i$ may be any i distinct elements of X and $(y^1, \ldots, y^i)$ may be any bit-string in $\{0,1\}^i$. We write SAMP_i to denote the set of all size-i samples, so $|\mathsf{SAMP}_i| = \binom{M}{i} \cdot 2^i \leq (2M)^i$. Observe that some elements of SAMP_i may not be consistent with any function $f \in \mathcal{C}$, while others may be consistent with many elements of $\mathcal{C}$.

The data structure DS consists of $\log N$ different "structures" which we refer to as $\mathcal{S}_0, \ldots, \mathcal{S}_{1+\log N}$. The i-th structure $\mathcal{S}_i$ is a set of (at most) $(2M)^i$ many "i-substructures" $\{\mathcal{S}_{i,\mathsf{samp}}\}_{\mathsf{samp} \in \mathsf{SAMP}_i}$ which are indexed by elements of SAMP_i. Given an element $\mathsf{samp} = \{(x^1, y^1), \ldots, (x^i, y^i)\} \in \mathsf{SAMP}_i$, an i-substructure $\mathcal{S}_{i,\mathsf{samp}}$ has two parts: the "main part" $\mathsf{MAIN}(\mathcal{S}_{i,\mathsf{samp}})$ and one additional function (which we explain and give notation for below). The main part $\mathsf{MAIN}(\mathcal{S}_{i,\mathsf{samp}})$ is the subset of $\mathcal{C}$ that contains precisely those concepts in $\mathcal{C}$ that are consistent with samp, i.e. those functions $f \in \mathcal{C}$ that have $f(x^j) = y^j$ for all $j \in [i]$. The one additional function is $\mathsf{MAJ}(\mathsf{MAIN}(\mathcal{S}_{i,\mathsf{samp}}))$, the function that outputs, on any input $x \in X$, the majority vote over all the concepts in $\mathsf{MAIN}(\mathcal{S}_{i,\mathsf{samp}})$. This concludes the description of a generic i-substructure $\mathcal{S}_{i,\mathsf{samp}}$, and thus concludes the description of the i-th structure $\mathcal{S}_i$.

For example, the zeroth structure $\mathcal{S}_0$ consists of only one 0-substructure (since there is only one "empty sample" with no labeled pairs); call this 0-substructure $\mathcal{S}_{0,\mathsf{samp}_\emptyset}$. We have $\mathsf{MAIN}(\mathcal{S}_{0,\mathsf{samp}_\emptyset}) = \mathcal{C}$ (since every concept is consistent with the empty sample) and the one additional function is just the first hypothesis that the halving algorithm uses, the majority vote across $\mathcal{C}$.

Construction of DS

Suppose that $\mathcal{S}_{i-1}$, the $(i-1)$-st structure, has been constructed by the pre-processing procedure. The structure $\mathcal{S}_i$ is built from $\mathcal{S}_{i-1}$ as follows. $\mathcal{S}_i$ has exactly $2^i\binom{M}{i}$ substructures, one for each possible samp of size i. Consider such a $\mathsf{samp} = \{(x^1, y^1), \ldots, (x^i, y^i)\}$, and let samp' be $\{(x^1, y^1), \ldots, (x^{i-1}, y^{i-1})\}$, its length-$(i-1)$ prefix. Since $\mathsf{MAIN}(\mathcal{S}_{i-1,\mathsf{samp}'})$ has been constructed already as part of $\mathcal{S}_{i-1}$, given x^i it is easy to enumerate over the functions in $\mathsf{MAIN}(\mathcal{S}_{i-1,\mathsf{samp}'})$ and partition them into two groups; the functions f for which $f(x^i) = 0$ will go into $\mathsf{MAIN}(\mathcal{S}_{i-1,\mathsf{samp}'\cup\{(x^i,0)\}})$ and the ones for which $f(x^i) = 1$ will go into $\mathsf{MAIN}(\mathcal{S}_{i-1,\mathsf{samp}'\cup\{(x^i,1)\}})$. Finally once we have $\mathsf{MAIN}(\mathcal{S}_{i-1,\mathsf{samp}'\cup\{(x^i,y^i)\}})$ it is straightforward to read the corresponding matrix and construct the one additional function $\mathsf{MAJ}(\mathsf{MAIN}(\mathcal{S}_{i-1,\mathsf{samp}'\cup\{(x^i,y^i)\}}))$. It is not hard to see that the total size of $\mathcal{S}_i$, and the total time required to build it from $\mathcal{S}_{i-1}$, is at most $(2M)^i \cdot O(NM)$. Even when $i = \log N$ this is at most $O(N \cdot M^{\log N} \cdot NM) = M^{O(\log N)}$.

Using DS to run the halving algorithm efficiently

Now we describe how to emulate the halving algorithm in total time $\mathrm{poly}(\log N, \log M)$ given access to the structures $\mathcal{S}_0, \ldots, \mathcal{S}_{\log N}$. The initial hypothesis of the halving algorithm is the additional function $\mathsf{MAJ}(\mathsf{MAIN}(\mathcal{S}_{0,\mathsf{samp}_\emptyset}))$, i.e. the majority vote over all functions in $\mathcal{C}$, so the emulator for the halving algorithm need only "point to" this portion of DS to construct its initial hypothesis. On receiving a first labeled counterexample (x^1, y^1), the hypothesis of the halving algorithm is then precisely $\mathsf{MAJ}(\mathsf{MAIN}(\mathcal{S}_{1,\{(x^1,y^1)\}}))$; conveniently, this has been pre-computed as part of $\mathcal{S}_1$, so the emulator again only needs to point to this portion of DS to construct its second hypothesis. On receiving a second labeled counterexample (x^2, y^2), the emulator updates its hypothesis by pointing to $\mathsf{MAJ}(\mathsf{MAIN}(\mathcal{S}_{2,\{(x^1,y^1),(x^2,y^2)\}}))$ as its hypothesis, and so on. This goes on for at most $\log N$ stages, and it is clear that each stage requires time at most $\mathrm{poly}(\log N, \log M)$, giving (ii) as claimed. ◀

4.3 The general learning result based on Nečiporuk's method

With Lemma 10 in hand, we are ready to state and prove our general learning result based on Nečiporuk's method. As suggested earlier, the approach is to first do random sampling to identify the "important" subfunctions (ones which occur in many subcubes), then run the pre-processing procedure using these important subfunctions and the halving algorithm to efficiently learn over all subcubes containing important subfunctions, patching up the hypothesis on any subcubes that do not contain important subfunctions.

▶ **Lemma 11** (Learning based on few subfunctions). *Fix any partition $S_1, S_2, \ldots, S_{n^{1-\delta}}$ of $[n]$ into equal-size subsets S_i of size n^δ each, where $\delta > 0$. Let $\mathcal{C}$ be a class of n-variable functions such that for each $f \in \mathcal{C}$ there is a set S_i such that $\mathrm{Num}(f, S_i) \leq 2^{n^\beta}$, where $\beta < 1$ and moreover $\beta + \delta < 1$. Then there is a randomized exact learning algorithm for $\mathcal{C}$ that uses membership and equivalence queries and achieves savings n^δ.*

Proof. We describe a randomized learning algorithm which works on S_1 and achieves the claimed runtime bound with high probability if $\mathrm{Num}(f, S_1) \leq 2^{n^\beta}$. If the algorithm runs for more than the claimed number of steps while working on S_1, it aborts and restarts, this time working on S_2, and so on. Hence in the following discussion we assume without loss of generality that $\mathrm{Num}(f, S_1) \leq 2^{n^\beta}$.

The learning algorithm works on S_1 in three stages as described below.

First stage: Identify important subfunctions

Recall that $\mathcal{R}_{S_1}$ is the set of all $2^{n-|S_1|} = 2^{n-n^\delta}$ restrictions that leave precisely the variables in S_1 free. Let $g_1, \ldots, g_{\mathrm{Num}(f,S_1)}$ be the subfunctions that occur in $\{f \restriction \rho\}_{\rho \in \mathcal{R}_{S_1}}$. For $i \in [\mathrm{Num}(f, S_1)]$ let p_i denote the fraction of the 2^{n-n^δ} subcubes in $\mathcal{R}_{S_1}$ that have g_i as the subcube there. We say that a subfunction g_i is *important* if $p_i \geq \varepsilon/(10 \cdot \mathrm{Num}(f, S_1))$ (we will specify the value of ε later). Let $F' \subseteq \{g_1, \ldots, g_{\mathrm{Num}(f,S_1)}\}$ be the set of all important subfunctions.

In the first stage we draw $A := 20n \cdot 2^{n^\beta}/\varepsilon$ independent uniform random elements of $\mathcal{R}_{S_1}$, and for each one we spend 2^{n^δ} many membership queries to exhaustively learn the associated truth table in time $\mathrm{poly}(n) \cdot 2^{n^\delta}$; let $F \subseteq \{g_1, \ldots, g_{\mathrm{Num}(f,S_1)}\}$ be the set of all the subfunctions that are discovered in this way. For any given fixed important subfunction, the probability that it is not included in F is at most $(1-\varepsilon/(10 \cdot \mathrm{Num}(f, S_1)))^A \leq (1-\varepsilon/(10 \cdot 2^{n^\beta}))^A < 1/2^{2n}$, so a union bound over all (at most 2^{n^β}) important subfunctions gives that with probability at least $1 - 1/2^n$ the set F contains the set F' of important subfunctions (for the rest of the algorithm's analysis and execution we suppose that indeed F contains F'). We observe that by the definition of F', at most an $\varepsilon/10$ fraction of all $\rho \in \mathcal{R}_{S_1}$ are such that $f \restriction \rho$ do not belong to F', and hence at most an $\varepsilon/10$ fraction of all $\rho \in \mathcal{R}_{S_1}$ are such that $f \restriction \rho$ do not belong to F.

Note that the total running time for the first stage of the algorithm is $2^{n^\delta} \cdot \mathrm{poly}(n) \cdot 2^{n^\beta}/\varepsilon$.

Second stage: Do the pre-processing on F

Next, the algorithm performs the pre-processing described in the previous subsection on the $N = |F|$ functions in F, each of which is defined over the $M = 2^{n^\delta}$-size domain $\{0,1\}^{S_1}$. Since $N \leq 2^{n^\delta}$ we have that this takes time at most $M^{O(\log N)} < 2^{O(n^\beta \cdot n^\delta)}$.

Third stage: Run the halving algorithm in parallel over all $Z := 2^{n-n^\delta}$ subcubes in $\mathcal{R}_{S_1}$ using the data structure DS from Lemma 10

In the amortized analysis of the halving algorithm with pre-processing given in the previous subsection, we assumed that every execution of the halving algorithm was performed on a target function that actually belonged to the class $\mathcal{C}$. In our current setting there may be up to $(\varepsilon/10) \cdot 2^{n-n^\delta}$ many subcubes that contain functions that are not in F. When the halving algorithm is run over a subcube that contains a subfunction in F, it will correctly converge to the target subfunction over that subcube after at most $\log N \leq n^\beta$ many counterexamples, and the efficient implementation of the halving algorithm via pre-processing will work correctly over that subcube. This will happen on at least $1 - \varepsilon/10$ fraction of all 2^{n-n^δ} subcubes. For the remaining (at most $(\varepsilon/10) \cdot 2^{n-n^\delta}$) subcubes that have a subfunction not on our list, the halving algorithm may not work correctly. If, in a given subcube, the version space ever vanishes (note that if this happens it must happen after at most $\log N$ counterexamples from that subcube), or it has size greater than one after $\log N$ counterexamples, then it must be the case that that subcube's subfunction does not belong to F. In this case the algorithm uses 2^{n^δ} membership queries to "patch up" the hypothesis over this subcube (which can be done in time $\mathrm{poly}(n) \cdot 2^{n^\delta}$). This happens for at most $(\varepsilon/10) \cdot 2^{n-n^\delta}$ subcubes. Thus, the total running time of this stage will be at most

$$
\begin{aligned}
&(\text{time on subcubes with subfunctions in } F) + (\text{time on other subcubes}) \\
&\leq Z \cdot \mathrm{poly}(\log N, \log M)) + (\varepsilon/10) \cdot 2^{n-n^\delta} \cdot \mathrm{poly}(n) \cdot 2^{n^\delta} \\
&\leq \mathrm{poly}(n) \cdot \left(2^{n-n^\delta} + (\varepsilon/10) \cdot 2^n\right).
\end{aligned}
$$

Hence the total running time for all stages can be upper bounded by

$$2^{n^\delta} \cdot \mathrm{poly}(n) \cdot 2^{n^\beta}/\varepsilon + 2^{O(n^\beta \cdot n^\delta)} + \mathrm{poly}(n) \cdot \left(2^{n-n^\delta} + (\varepsilon/10) \cdot 2^n\right).$$

Recalling that by assumption $\beta + \delta < 1$, we may take $\varepsilon = 2^{-n/2}$, and the overall running time is at most $\mathrm{poly}(n) \cdot 2^{n-n^\delta}$. ◀

4.4 Instantiating Lemma 11 using Nečiporuk's lower bounds

The proof of Corollary 9 extends unchanged to give our concrete learning results based on Nečiporuk's lower bounds:

▶ **Corollary 12.** *There is a randomized exact learning algorithm, using membership and equivalence queries, to learn Boolean n-variable functions computable by computational model $\mathcal{A}$ of size S in time 2^{n-n^ε} for a fixed $\varepsilon > 0$, where $\mathcal{A}$ and S can be instantiated as in items (1)–(6) of Corollary 9.*

5 Conclusion

We initiated the study of learning algorithms with non-trivial savings and gave a range of such learning algorithms for various natural circuit classes. There are many intriguing problems left open by our work, we list a few.

- Our learning algorithms of Sections 3 and 4 are based on influential lower bound techniques in circuit complexity, namely the method of random restrictions and Nečiporuk's lower bound method. Can other proof techniques from circuit complexity, such as Razborov's method of approximations for monotone circuit lower bounds [35, 36] or the "polynomial method" for various classes of constant-depth circuits [5], similarly be leveraged to obtain non-trivial learning algorithms?
- Related to the previous item, there are several prominent circuit classes for which lower bounds are known but we do not yet have non-trivial learning algorithms; examples include intersections of $\mathrm{poly}(n)$ many LTFs, de Morgan formulas of size $n^{2.99}$, monotone formulas of significantly sublinear depth, monotone circuits of polynomial size, etc. Can we develop learning algorithms with nontrivial savings for these classes?
- We strongly suspect that many of our learning results, specifically the ones for AC^0, LTF-of-AC^0, parity-of-AC^0, and all the classes covered by Corollary 12, are best possible given the state of the art of circuit lower bounds. If we had *deterministic* learning algorithms then this would follow from the work of [28] (see their Theorem 1). Is it possible to design deterministic algorithms for these classes (or alternatively, to extend Theorem 1 of [28] to cover randomized learning algorithms)?
- Finally, we close with an ambitious twist on the previous bullet: in the spirit of recent celebrated work [43, 44] wringing exciting new circuit lower bounds from non-trivial satisfiability algorithms, is it possible to leverage ideas from non-trivial learning algorithms to obtain new circuit lower bounds?

References

1 D. Angluin. Learning Regular Sets from Queries and Counterexamples. *Information and Computation*, 75(2):87–106, 1987.

2 D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

3 Ya M Barzdin and RV Freivald. Prediction of general recursive functions. *Doklady Akademii Nauk SSSR*, 206(3):521, 1972.

4 P. Beame, R. Impagliazzo, and S. Srinivasan. Approximating AC^0 by Small Height Decision Trees and a Deterministic Algorithm for $\#AC^0$-SAT. In *CCC*, pages 117–125, 2012.

5 R. Beigel. The polynomial method in circuit complexity. In *Proceedings of the Eigth Conference on Structure in Complexity Theory*, pages 82–95, 1993.

6 A. Beimel, F. Bergadano, N. Bshouty, E. Kushilevitz, and S. Varricchio. On the applications of multiplicity automata in learning. In *Proceedings of the Thirty-Seventh Annual Symposium on Foundations of Computer Science*, pages 349–358, 1996.

7 Andreas Björklund. Counting perfect matchings as fast as Ryser. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 914–921, 2012.

8 Avrim Blum. Separating distribution-free and mistake-bound learning models over the boolean domain. *SIAM J. Comput.*, 23(5):990–1000, 1994.

9 A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

10 N. Bshouty. Exact learning via the monotone theory. *Information and Computation*, 123(1):146–153, 1995.

11 Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning Algorithms from Natural Proofs. In Ran Raz, editor, *31st Conference on Computational Complexity (CCC 2016)*, volume 50 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:24, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:http://dx.doi.org/10.4230/LIPIcs.CCC.2016.10.

12 Ruiwen Chen, Valentine Kabanets, Antonina Kolokolova, Ronen Shaltiel, and David Zuckerman. Mining circuit lower bound proofs for meta-algorithms. *Computational Complexity*, 24(2):333–392, 2015.

13 P. Fischer and H. Simon. On learning ring-sum expansions. *SIAM Journal on Computing*, 21(1):181–192, 1992.

14 Fedor V Fomin and Dieter Kratsch. *Exact exponential algorithms*. Springer Science & Business Media, 2010.

15 P. Gopalan and R. Servedio. Learning and lower bounds for AC^0 with threshold gates. In *Proc. 14th Intl. Workshop on Randomization and Computation (RANDOM)*, pages 588–601, 2010.

16 Parikshit Gopalan, Adam R. Klivans, and Raghu Meka. Learning functions of halfspaces using prefix covers. *Journal of Machine Learning Research - Proceedings Track*, 23:15.1–15.10, 2012.

17 Johan Håstad. *Computational Limitations for Small Depth Circuits*. MIT Press, Cambridge, MA, 1986.

18 Johan Håstad. On the correlation of parity and small-depth circuits. *SIAM Journal on Computing*, 43(5):1699–1708, 2014.

19 L. Hellerstein and R. Servedio. On PAC learning algorithms for rich boolean function classes. *Theoretical Computer Science*, 384(1):66–76, 2007.

20 D. Helmbold, R. Sloan, and M. Warmuth. Learning integer lattices. *SIAM Journal on Computing*, 21(2):240–266, 1992.

21 Russell Impagliazzo, William Matthews, and Ramamohan Paturi. A satisfiability algorithm for AC^0. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 961–972, 2012.

22 Taisuke Izumi and Tadashi Wadayama. A new direction for counting perfect matchings. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 591–598, 2012.

23 Stasys Jukna. *Boolean Function Complexity*. Springer, 2012.

24 Daniel M. Kane. The correct exponent for the gotsman-linial conjecture. In *Proc. 28th Annual IEEE Conference on Computational Complexity (CCC)*, 2013.

25 M. Karchmer and A. Wigderson. On span programs. In *Proceedings of the Eighth Annual Structure in Complexity Theory Conference (San Diego, CA, 1993)*, pages 102–111. IEEE Comput. Soc. Press, Los Alamitos, CA, 1993. doi:10.1109/SCT.1993.336536.

26 A. Klivans, R. O'Donnell, and R. Servedio. Learning intersections and thresholds of halfspaces. *Journal of Computer & System Sciences*, 68(4):808–840, 2004.

27 A. Klivans and R. Servedio. Learning DNF in time $2^{\tilde{O}(n^{1/3})}$. *Journal of Computer & System Sciences*, 68(2):303–318, 2004.

28 Adam Klivans, Pravesh Kothari, and Igor Carboni Oliveira. Constructing hard functions using learning algorithms. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013*, pages 86–97, 2013.

29 Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

30 N. Littlestone. Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

31 W. Maass and G. Turan. How fast can a threshold gate learn? In *Computational Learning Theory and Natural Learning Systems: Volume I: Constraints and Prospects*, pages 381–414. MIT Press, 1994.

32 Ramamohan Paturi, Pavel Pudlák, Michael E. Saks, and Francis Zane. An improved exponential-time algorithm for k-SAT. *J. ACM*, 52(3):337–364 (electronic), 2005. doi:10.1145/1066100.1066101.

33 Ramamohan Paturi, Pavel Pudlák, and Francis Zane. Satisfiability coding lemma. *Chicago J. Theoret. Comput. Sci.*, pages Article 11, 19 pp. (electronic), 1999.

34 Pavel Pudlák. The hierarchy of boolean circuits. *Computers and artificial intelligence*, 6(5):449–468, 1987.

35 A. Razborov. Lower bounds on the monotone complexity of some boolean functions. *Dokl. Akad. Nauk SSSR*, 281:798–801, 1985. English translation in: Soviet Math. Dokl. 31:354–357, 1985.

36 Alexander A. Razborov. On the method of approximations. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 167–176, 1989.

37 Ben W. Reichardt. Reflections for quantum query algorithms. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 560–569. SIAM, Philadelphia, PA, 2011.

38 R. Rivest. Learning decision lists. *Machine Learning*, 2(3):229–246, 1987.

39 T Schoning. A probabilistic algorithm for k-sat and constraint satisfaction problems. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 410–414. IEEE, 1999.

40 Rainer Schuler. An algorithm for the satisfiability problem of formulas in conjunctive normal form. *J. Algorithms*, 54(1):40–44, 2005.

41 A. Tal. Tight Bounds on The Fourier Spectrum of AC^0. ECCC report TR14-174 Revision #1, 2015. URL: http://eccc.hpi-web.de/report/2014/174/.

42 L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

43 Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. In *STOC'10—Proceedings of the 2010 ACM International Symposium on Theory of Computing*, pages 231–240. ACM, New York, 2010.

44 Ryan Williams. Non-uniform ACC circuit lower bounds. In *26th Annual IEEE Conference on Computational Complexity*, pages 115–125. IEEE Computer Soc., Los Alamitos, CA, 2011.

45 Virginia V. Williams. Hardness of easy problems: basing hardness on popular conjectures such as the Strong Exponential Time Hypothesis. In *10th International Symposium on Parameterized and Exact Computation*, volume 43 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 17–29. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2015.

A Proof of Corollary 5

Proof. We shall assume that the depth-$(d+1)$ circuit F is *layered*, meaning that for any gate g it contains, every directed path from an input variable to g has the same length (converting an unlayered circuit to a layered one increases its size only by a factor of d). We prove the corollary with a failure probability of $S \cdot 2^{-t}$ for such layered circuits. Let S_i denote the number of gates in layer i (at distance i from the inputs), so $S = S_1 + \cdots + S_d$.

We begin by trimming the bottom fan-in of F: applying Theorem 4 with $\mathcal{F}$ being the S_1 many bottom layer gates of F (viewed as depth-2 circuits of bottom fan-in $w = 1$) and $p_0 := 1/48$, we get that

$$\begin{aligned}&\mathop{\mathbf{Pr}}_{\boldsymbol{\rho} \leftarrow \mathcal{R}_{p_0}} \left[F \restriction \boldsymbol{\rho} \text{ is not a } (t, G \circ \mathsf{AC}^0(\text{depth } d, \text{ bottom fan-in } \log S))\text{-decision tree} \right] \\ &\leq S_1 \cdot 2^{-t}.\end{aligned}$$

Let $F^{(0)}$ be any good outcome of the above, a $(t, G \circ \mathsf{AC}^0(\text{depth } d, \text{ bottom fan-in } \log S))$-decision tree. Note that there are at most 2^t many $\mathsf{AC}^0(\text{depth } d, \text{ fan-in } \log S)$ circuits at the leaves of the depth-t decision tree. Applying Theorem 4 to each of them with $p_1 := 1/(48 \log S)$ (and the 't' of Theorem 4 being $2t$) and taking a union bound over all 2^t many of them, we get that

$$\begin{aligned}&\mathop{\mathbf{Pr}}_{\boldsymbol{\rho} \leftarrow \mathcal{R}_{p_1}} \left[F^{(0)} \restriction \boldsymbol{\rho} \text{ is not a } (t + 2t, G \circ \mathsf{AC}^0(\text{depth } d - 1, \text{ fan-in } \log S))\text{-decision tree} \right] \\ &\leq S_2 \cdot 2^{-2t} \cdot 2^t = S_2 \cdot 2^{-t}.\end{aligned}$$

Repeat with $p_2 = \ldots = p_{d-1} := 1/(48 \log S)$, each time invoking Theorem 4 with its 't' being the one more than the current depth of the decision tree, so at the j-th invocation Theorem 4 is invoked with its 't' being 2^{j-1}. The claim then follows by summing the $S_1 2^{-t}$, $S_2 2^{-t}, \ldots, S_d 2^{-t}$ failure probabilities over all d stages and the fact that

$$\prod_{j=0}^{d-1} p_i = \frac{1}{48} \cdot \frac{1}{(48 \log S)^{d-1}}.$$ ◀

Expander Construction in VNC^1

Sam Buss[*1], **Valentine Kabanets**[† 2], **Antonina Kolokolova**[‡3], **and Michal Koucký**[§4]

1 **Department of Mathematics,University of California San Diego, La Jolla, USA**
sbuss@ucsd.edu
2 **School of Computing Science, Simon Fraser University, Burnaby, Canada**
kabanets@cs.sfu.ca
3 **Department of Computer Science, Memorial University of Newfoundland, St. John's, Canada**
kol@cs.mun.ca
4 **Computer Science Institute, Charles University, Prague, Czech Republic**
koucky@iuuk.mff.cuni.cz

Abstract

We give a combinatorial analysis (using edge expansion) of a variant of the iterative expander construction due to Reingold, Vadhan, and Wigderson [38], and show that this analysis can be formalized in the bounded arithmetic system VNC^1 (corresponding to the "NC^1 reasoning"). As a corollary, we prove the assumption made by Jeřábek [24] that a construction of certain bipartite expander graphs can be formalized in VNC^1. This in turn implies that every proof in Gentzen's sequent calculus LK of a monotone sequent can be simulated in the monotone version of LK (MLK) with only polynomial blowup in proof size, strengthening the quasipolynomial simulation result of Atserias, Galesi, and Pudlák [7].

1998 ACM Subject Classification F.4.1 Mathematical Logic, F.2.1 Numerical Algorithms and Problems, F.1.3 Complexity Measures and Classes

Keywords and phrases expander graphs, bounded arithmetic, alternating log time, sequent calculus, monotone propositional logic

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.31

1 Introduction

Expander graphs have become one of the most useful combinatorial objects in theoretical computer science, with many beautiful applications in computer science and mathematics [19], and responsible for several breakthroughs in computational complexity [37, 17]. These graphs have seemingly contradictory properties: sparseness and high connectivity. The high connectivity can be measured in a number of different, but essentially equivalent ways: vertex expansion (every small subset of vertices "expands", i.e., has a larger neighborhood), edge expansion (every small subset of vertices has many edges leaving the set), or fast mixing time (a random walk on a regular expander graph quickly converges to the uniform distribution on vertices).

* Supported in part by NSF grant CCF-1213151. Part of the work on VNC^1 was done while Buss was visiting the Chebychev Laboratory, St.Petersburg State University in Spring 2016, supported in part by Skolkovo Institute of Science and Technology.
† Supported in part by an NSERC Discovery grant. Part of the work was done while visiting UCSD.
‡ Supported in part by an NSERC Discovery grant. Part of the work was done while visiting UCSD.
§ The research leading to these results has received funding from the European Research Council under the European Unions Seventh Framework Programme (FP/2007-2013)/ERC Grant Agreement n. 616787.


© Sam Buss, Valentine Kabanets, Antonina Kolokolova, and Michal Koucký;
licensed under Creative Commons License CC-BY

The existence of expander graphs of constant degree can be argued nonconstructively using a simple probabilistic argument: for any constant $d \geq 3$, a random d-regular graph is almost surely an expander [33]. Constructing such graphs efficiently deterministically is much more difficult. The first explicit constructions were given by Margulis [27] and Gabber and Galil [18]. Lubotzky, Phillips, and Sarnak [25] gave a construction of expanders with particularly interesting properties, called Ramanujan graphs. All of these constructions are algebraic in nature: a graph is defined using a certain algebraic object (e.g., a group). Moreover, the analysis of correctness of the constructions is also algebraic. It relies on the algebraic notion of high connectivity called the *eigenvalue gap* and defined as follows. Consider the adjacency matrix of a given undirected d-regular graph, compute its eigenvalues, and order them according to the absolute value. It can be easily checked that d is the largest value. The difference between d and (the absolute value of) the second largest eigenvalue is the eigenvalue gap. The bigger this eigenvalue gap, the more connected the graph is. From this point of view, a d-regular expander is a graph with the eigenvalue gap at least $\Omega(d)$, i.e., the second largest eigenvalue should be at most some constant fraction of the degree.

A simpler, fully combinatorial construction of constant-degree expanders was given by Reingold, Vadhan, and Wigderson [38]. They started with constant-size expander graphs (which can found by brute-force search), and iteratively applied certain graph operations that increase the size of the graph while preserving its expansion property. This way, one can quickly construct an expander graph of any given size. While the construction of [38] is combinatorial, its analysis is still algebraic and is based on estimating the eigenvalue gap. Alon, Schwartz, and Shapira [4] gave a different construction of expanders, which combines algebraically constructed expanders of Alon and Roichman [3] with only two applications of a certain graph operation (replacement product), to obtain a constant-degree expander of arbitrary size. They also gave a fully combinatorial analysis of the replacement product operation they used in the second stage of the construction. Their full analysis, however, is still algebraic, as it relies on the algebraic construction and the eigenvalue gap analysis of [3]. In this respect, the situation in [4] is similar to that in [38] where the analysis of a related graph operation (zig-zag product) can be done in terms of min-entropy, while the analysis of the complete construction is still based on eigenvalues.

The focus of our paper is to give a construction of expanders with a simple (non-algebraic) analysis, where simplicity is measured in terms of the power of a system of bounded arithmetic needed to formalize the analysis. Informally, systems of bounded arithmetic are obtained by restricting the power of the standard first-order theory of Peano arithmetic. It is possible to devise systems of bounded arithmetic that correspond to systems of reasoning using only concepts from a given complexity class, e.g., P or NC^1. A natural question is: what is the weakest complexity class so that the existence of expander graphs can be proved using only the concepts of that complexity class?

The known expander constructions mentioned above can be formalized within a system of polytime reasoning, intuitively because eigenvalues and matrix determinants are known to be computable in polytime. Our main result is a construction of expanders that can be formalized within a system of NC^1 reasoning, VNC^1 (see below for a formal definition). As NC^1 algorithms are not known to compute the eigenvalues or determinant of a given matrix, any such formalization of an expander construction in VNC^1 must necessarily avoid the use of eigenvalues, and hence be "combinatorial" in that sense.

As expanders are used in a number of complexity-theoretic results, formalizing the expander construction within a weak system of bounded arithmetic is an important step in formalizing these complexity-theoretic results within the bounded arithmetic framework, which in turn may have other implications. For example, in proof complexity, we can use

our expander construction to argue that any Gentzen's sequent calculus LK proof (of a monotone sequent) can be simulated by a *monotone* LK (MLK) proof, with only polynomial blowup in proof size, improving upon the quasipolynomial simulation shown by Atserias, Galesi and Pudlák [7], and answering a question of Pudlák and Buss [36]. This simulation result follows by the work of Jeřábek [24] who proved the result under the assumption that a certain expander graph family can be proved to exist within a system of NC^1 reasoning. Our paper proves a strengthening of the assumption needed by Jeřábek.

1.1 Our results

Our main contribution is the analysis of one of the iterative expander constructions from [38], which we show to be formalizable in the bounded arithmetic system VNC^1 (of NC^1 reasoning). As in [38], the expander construction is *fully explicit* in the sense that that there is a deterministic polynomial-time algorithm that, given a vertex name v in binary and a number i, outputs the value of the rotation map $Rot(v,i) = (w,j)$, where w is the name of the ith neighbor of v in the graph, and j is the number such that v is the jth neighbor of w. Moreover, we show that there is an alternating linear-time algorithm that accepts exactly the triples of the form $\langle v, i, Rot(v,i)\rangle$; this kind of explicitness is what we will use to argue that the expander construction is formalizable in VNC^1.

▶ **Theorem 1** (Main result: Informal version). *The existence of an expander graph family can be proved using* NC^1 *reasoning only (within the system* VNC^1*).*

As our main application, building on Jeřábek [24] and Atserias, Galesi and Pudlák [7], we show that every proof in Gentzen's sequent calculus LK of a monotone sequent can be simulated by a monotone LK (MLK) proof (a sequent calculus proof in which all formulas are positive) with only polynomial blowup in size. This answers a question of Pudlák and Buss [36]. Previously, [7] showed such simulation with quasipolynomial blowup in proof size.

▶ **Theorem 2** (Main application). MLK *polynomially simulates* LK *on monotone sequents.*

It is easy to show that the intuitionistic propositional sequent calculus LJ polynomially simulates MLK (see Pudlák [34] and Bilková [8]); thus we get as an immediate corollary that propositional LJ polynomially simulates LK on monotone sequents, re-proving the result of Jeřábek [22, Theorem 3.9]. Many of the principles that have been considered in propositional proof complexity are expressed as monotone sequents, notably the pigeonhole principle and the clique-coloring tautologies. As these principles have polynomial size LK proofs [11], Theorem 2 implies that they also have polynomial size proofs in MLK as well as in propositional LJ. The prior best known results for the pigeonhole principle were the quasipolynomial size MLK proofs of Atserias, Galesi and Gavaldà [6].

It remains an open problem whether tree-like MLK can polynomially simulate MLK, equivalently whether tree-like MLK can polynomially simulate LK on monotone sequents. Note that [7] gives a quasipolynomial simulation.

Intuitively, to simulate an LK proof within MLK, one needs to construct (and prove correctness of) a monotone formula for the majority function. Such monotone formulas can be built using the classical AKS sorting networks [1]. Jeřábek [24] shows that the analysis of AKS sorting networks can be formalized within a certain system of NC^1 reasoning (slightly more powerful than VNC^1), under the assumption that the existence of expander graphs, with certain parameters, is also formalizable within the same system. Our Theorem 1 proves the assumption needed by Jeřábek (actually a slightly stronger version, as our proof of the existence of expanders is in the weaker system VNC^1), and so Theorem 2 immediately follows.

1.2 Relation to previous work

1.2.1 Expander constructions

The expander graph construction that we analyze is a variant of the iterative construction of expanders given in [38]. The idea is to start with a constant-size expander graph (found, say, by exhaustive search), and iteratively increase the size of the graph while keeping its expansion larger than some universal constant. The notion of expansion used by [38] is in terms of the eigenvalue gap. To analyze the expansion of the final graph, Reingold, Vadhan, and Wigderson [38] bound the effect of the graph operations they used (graph powering, graph tensoring, and zig-zag product) on the second largest eigenvalue of the adjacency matrix of the resulting graph. The analysis of graph powering (where an edge of the kth power of a graph G is a walk of length k in G) and graph tensoring (where an edge of the tensor product of G and H consists of a pair of edges, one from G and one from H) is immediate from the basic linear algebra. The analysis of the zig-zag product (a way to compose a graph G with a graph H so that the new graph has the degree of H) is technically the most difficult part of the algebraic analysis of the expander construction in [38].

In [4], a graph replacement product (closely related to the zig-zag product) is analyzed in terms of edge expansion, avoiding any mention of the eigenvalue gap. Since replacement product can be used instead of zig-zag product in an iterative expander construction along the lines of [38], this gives a combinatorial analysis of the part of the expander construction. In order to make the entire analysis combinatorial, it suffices to analyze graph powering and graph tensoring also in terms of edge expansion. This is exactly what we do in the present paper.

Our combinatorial analysis of graph tensoring, though subtle, is not very difficult. For the analysis to go through, it turns out necessary to work with graphs that have sufficiently many self-loops around every vertex (at least half the degree). On the other hand, graph powering is much more difficult to analyze combinatorially. Fortunately, here we were able to use the result of Mihail [28] who gave a combinatorial analysis of the mixing time of random walks on expanders in terms of edge expansion. (Interestingly, for her proof, she also had to work with graphs that have many self-loops around every vertex.) Finally, using Mihail's bounds, we are able to conclude the analysis of graph powering in terms of edge expansion, borrowing some ideas from [2].

1.2.2 Bounded arithmetic

There is a long history of formalizing complexity results in bounded arithmetic; indeed, this was one of the main motivations for the definitions of bounded arithmetic. First, bounded arithmetic theories can capture a range of complexity classes, from uniform AC^0 and uniform NC^1, to polynomial time, polynomial space and exponential time (see [10, 14]). Second, via the Paris-Wilkie or Cook translations, proofs in bounded arithmetic can be viewed as uniform families of propositional proofs. For this reason, a proof in bounded arithmetic can sometimes yield new propositional proofs.

There has been considerable progress in formalizing advanced results from computational complexity in weak theories of bounded arithmetic; these include approximate counting, randomized computations, and Arthur-Merlin games [20, 21], Toda's theorem [12], and the PCP theorem [32]. The present paper continues this tradition by formalizing the construction of expander graphs in the weak fragment VNC^1 which corresponds to NC^1 computation.

There are a number of prior works which use bounded arithmetic to obtain upper bounds in proof complexity. A big advantage of using bounded arithmetic is that the proofs can

be considerably simplified. A classic example is the work by Paris and Wilkie [29] who showed that the proofs of the weak pigeonhole principle in $I\Delta_0$ constructed by [30] yield constant-depth, polynomial-size Frege proofs of the propositional translations of the weak pigeonhole principle (via the "Paris-Wilkie translation"). Lower-depth, quasipolynomial-size Frege proofs were later given by [26] via a proof of the weak pigeonhole principle in a different fragment of bounded arithmetic. Similarly, [35] gave proofs of Ramsey's theorem in S_2, and these translate into quasipolynomial-size, constant-depth Frege proofs. Recently, [12] used formalization of Toda's theorem in bounded arithmetic with modular counting quantifiers to show that constant-depth $\mathsf{AC}^0(p)$-proofs, for p a prime, can be translated into quasipolynomial size, depth-three propositional proofs, with formulas being Boolean combinations of mod p gates of small conjunctions. Another classic example is Cook's theorem that extended Frege proofs have polynomial size proofs of their partial consistency statements, which was established via provability in PV [15].

The present paper establishes a new result of this type via a Cook-style translation: together with earlier work of Jeřábek [23], our formalization of expander graphs in VNC^1 implies that the monotone propositional proof system MLK polynomially simulates the proof system LK. We will use the system VNC^1 defined by Cook and Morioka [16]. We conservatively extend VNC^1 to facilitate reasoning about the compositions of NC^1 functions, which allows us to simplify the formalization of our recursive expander construction.

1.2.2.1 Remainder of the paper

Section 2 contains basic definitions. Our expander construction is defined in Section 3. In Section 4, we present a construction of bipartite expanders needed by Jeřábek [24]. In Section 5, we show that the existence of our expander graphs is provable in VNC^1, thereby proving a formal version of Theorem 1. We derive Theorem 2 in Section 6. Section 7 contains concluding remarks. For space considerations, some proofs are only sketched and other proofs are omitted from this conference version; for the full version (with all missing proofs), please see [9].

2 Preliminaries

2.1 Notation

We consider undirected graphs, possibly with parallel edges and self-loops. For an undirected graph $G = (V, E)$ on n nodes, we usually associate the vertex set V with the set $[n] = \{1, 2, \ldots, n\}$, and denote an edge $i \sim j$ between nodes i and j as $\{i, j\} \in E$. In this notation, we also allow self-loops $\{i, i\} \in E$.

The adjacencies of a d-regular graph G are given via its rotation map Rot_G so that, for vertex v of G and an index $i \in [d]$, we have $Rot_G(v, i) = (w, j)$ if w is the ith neighbor of v, and v is the jth neighbor of w; so, in particular, the rotation map induces some fixed numbering of neighbors of a given vertex.

For an n-vertex graph G, its *adjacency matrix* is an $n \times n$ matrix A' whose (i, j)th entry contains the number of edges between vertices i and j in G. For d-regular graphs G, it will be more convenient for us to consider the *normalized adjacency matrix* defined as $\frac{1}{d} \cdot A'$. Note that the normalized adjacency matrix A of G is the probability transition matrix for a random walk on G. That is, if π is a probability distribution on vertices of G, then $A\pi$ is the probability distribution induced by one step of a random walk on G starting from a vertex distributed according to π. It is also easy to see that A^k is the normalized adjacency matrix of the graph G^k.

2.2 Expanders

For a graph $G = (V, E)$ and a subset $U \subseteq V$ of vertices, we denote by $\overline{U}$ the set $V \setminus U$, and by $E(U, \overline{U})$ the set of edges between U and $\overline{U}$. The *edge expansion* of a d-regular graph $G = (V, E)$ on n vertices is defined as

$$\min_{\emptyset \neq U \subset V,\ |U| \leq n/2} \frac{|E(U,\overline{U})|}{d \cdot |U|} = \min_{\emptyset \neq U \subset V} \frac{|E(U,\overline{U})|}{d \cdot \min\{|U|, |\overline{U}|\}}. \tag{1}$$

For a graph $G = (V, E)$ and a subset $U \subseteq V$ of vertices, we denote by $\Gamma_G(U)$ the set of all neighbors of U in G, i.e.,

$$\Gamma_G(U) = \{v \in V \mid \exists u \in U, \{u, v\} \in E\}.$$

We drop the subscript G if the graph G is understood from the context. We denote by $\Gamma^+(U)$ the set $\Gamma(U) \setminus U$ of new neighbors of U. The *vertex expansion* of a graph $G = (V, E)$ on n vertices is defined as

$$\min_{\emptyset \neq U \subset V, |U| \leq n/2} \frac{|\Gamma^+(U)|}{|U|}.$$

2.3 Bounded arithmetic theory VNC^1

A number of bounded arithmetic theories have been proposed for uniform NC^1: these include the theory $\mathrm{A}^{\log}$ of Clote and Takeuti [13], the theory AID of Arai [5], the theory VNC^1 of Cook and Morioka [16], and a reformulated version of VNC^1 by Cook and Nguyen [14]. Jeřábek [23] describes a theory VNC^1_* for NC^1 under a relaxed notion of uniformity for logarithmic depth circuits.

Cook and Morioka [16] define VNC^1 using tree recursion (*TreeRec*). Cook and Nguyen [14] give an equivalent definition of VNC^1 using the Boolean formula value problem. It is easier to formalize the expander graph construction with tree recursion, so we work with the version of VNC^1 as defined by Cook and Morioka [16].

The bounded arithmetic theory VNC^1 is an extension of the theory V^0 of bounded arithmetic; V^0 corresponds in power to AC^0. V^0 is a second-order (two-sorted) system of arithmetic, with two sorts of numbers (first-order objects) and strings (second order objects). Strings are viewed as members of $\{0,1\}^*$. The notation $X(i)$, where X is a string and $i \geq 0$ is a natural number, means the Boolean value of the i^{th} entry in string X. Sometimes $i{\in}X$ is written instead of $X(i)$. The constants 0 and 1 are number terms, and addition and multiplication are number functions. Another term of type number is string length $|X|$, defined to be the value of the largest element in X when viewed as a set plus 1. Addition and multiplication are defined for numbers only, and equality is defined both for numbers and strings. The axioms of V^0 consist of a finite set of "BASIC" open axioms defining simple properties of the constants, relation symbols and function symbols, plus Σ^B_0-Comprehension axioms

$$\Sigma^B_0\text{-COMP:} \quad \exists X{\leq}y\, \forall z{<}y\, (X(z) \leftrightarrow \varphi(z))$$

for any formula φ in Δ^B_0 not containing X as a free variable, but possibly containing free variables other than z. A Δ^B_0 formula is one in which all quantifiers are bounded and which contains no second-order quantifiers. We write $(\exists X{\leq}y)\psi$ for $\exists X((|X|{\leq}y) \wedge \psi)$.

Let $\varphi(i, \vec{x}, \vec{X})[p, q]$ and $\psi(i, \vec{x}, \vec{X})$ be Σ^B_0-formulas. The notation "$[p, q]$" indicates that p and q are propositional variables that may occur as atomic subformulas in φ. The Σ^B_0-*TreeRec*

property [16] is defined by the formula $B^{\varphi,\psi}(a,\vec{x},\vec{X},Z)$:

$$(\forall i{<}a)[(Z(a{+}i) \leftrightarrow \psi(i)) \wedge (Z(i) \leftrightarrow \varphi(i,\vec{x},\vec{X})[Z(2i{+}1), Z(2i{+}2)])].$$

For $i < a$, this states that $Z(i)$ is a Boolean function of the two values $Z(2i+1)$ and $Z(2i+2)$. Thus $Z(i)$ is computed by a circuit which is a formed as a binary tree with gate types specified by φ and input values specified by ψ. We can always assume w.l.o.g. that $a = 2^{|a|}-1$; we call this the "depth condition" and it means the binary tree is exactly balanced and of depth $|a|$. This tree is of course a fanin two Boolean circuit. The type of the i-th gate is determined by $\varphi(i,\vec{x},\vec{X})$ and thus is a Σ^B_0-property of i and the inputs $\vec{x}$ and $\vec{X}$.

The theory VNC^1 is defined as V^0 plus the Σ^B_0-*TreeRec* axioms $(\exists Z{\leq}2a)B^{\varphi,\psi}(a,\vec{x},\vec{X},Z)$ for all φ and ψ in Σ^B_0. The language of VNC^1 can be extended by adding a new relation symbol $R^{\varphi,\psi}(i,a,\vec{x},\vec{X})$ for every formula $B^{\varphi,\psi}$. The defining axioms for $R^{\varphi,\psi}$ are

$$B^{\varphi,\psi}(a,\vec{x},\vec{X},R^{\varphi,\psi}) \qquad \text{and} \qquad i{\geq}2a \rightarrow \neg R^{\varphi,\psi}(i,a,\vec{x},\vec{X}).$$

Note that the defining axioms uniquely specify all the values of $R^{\varphi,\psi}$, provably in VNC^1. Adding the predicate symbols $R^{\varphi,\psi}$ and their defining axioms to VNC^1 yields the theory $\mathsf{VNC}^1(\mathit{TreeRec})$.[1] As an extension by definitions, this theory is conservative over VNC^1. This means that VNC^1 and $\mathsf{VNC}^1(\mathit{TreeRec})$ can be used interchangeably. Indeed, any $\forall\Sigma^B_1(\mathit{TreeRec})$-formula which is provable in $\mathsf{VNC}^1(\mathit{TreeRec})$ can be translated naturally to an equivalent $\forall\Sigma^B_1$-formula which is VNC^1-provable. Thus, in Section 5, we may work in VNC^1 but still use the full power of $\mathsf{VNC}^1(\mathit{TreeRec})$.

A key property of VNC^1 is that it can Σ^B_1-define precisely the (uniform) NC^1 functions; this is discussed in Section 5.1.

2.4 LK and MLK proof systems

The system MLK of monotone reasoning in [7] is a variant of Gentzen's sequent calculus LK in which all formulas are positive. An LK proof is a list of sequents of the form $\varphi_1,\ldots,\varphi_n \rightarrow \psi_1,\ldots\psi_m$, interpreted as $\bigwedge_{i=1}^n \varphi_i \rightarrow \bigvee_{j=1}^m \psi_j$. The axioms are $\varphi \rightarrow \varphi$, $\Gamma \rightarrow 1$, and $0 \rightarrow \Gamma$, for an arbitrary list of formulas Γ. Let φ,ψ denote formulas and Γ,Δ lists of formulas. The main derivation rules of LK are as follows.

- **Left derivation:** $\dfrac{\varphi,\psi,\Gamma\rightarrow\Delta}{(\varphi\wedge\psi),\Gamma\rightarrow\Delta}$ $\quad$ $\dfrac{\varphi,\Gamma\rightarrow\Delta \qquad \psi,\Gamma'\rightarrow\Delta'}{(\varphi\vee\psi),\Gamma,\Gamma'\rightarrow\Delta,\Delta'}$ $\quad$ $\dfrac{\Gamma\rightarrow\varphi,\Delta}{\neg\varphi,\Gamma\rightarrow\Delta}$
- **Right derivation:** $\dfrac{\Gamma\rightarrow\Delta,\varphi,\psi}{\Gamma\rightarrow\Delta,(\varphi\vee\psi)}$ $\quad$ $\dfrac{\Gamma\rightarrow\Delta,\varphi \qquad \Gamma'\rightarrow\Delta',\psi}{\Gamma,\Gamma'\rightarrow\Delta,\Delta',(\varphi\wedge\psi)}$ $\quad$ $\dfrac{\varphi,\Gamma\rightarrow\Delta}{\Gamma\rightarrow\Delta,\neg\varphi}$
- **Cut rule:** $\dfrac{\Gamma\rightarrow\Delta,\varphi \qquad \varphi,\Gamma'\rightarrow\Delta'}{\Gamma,\Gamma'\rightarrow\Delta,\Delta'}$

Additionally, LK includes structural rules on both sides of a sequent such as weakening, contraction of duplicate formulas, and changing order of formulas on the same side. LK is equivalent in power to Frege systems, and tree-like LK is equivalent to LK, thus VNC^1 proofs translate into polynomial-size LK proofs [5, 16, 14].

In Monotone LK (MLK), all formulas in the proof are over the $\wedge,\vee$ basis with no $\neg$.

[1] Cook and Morika [16] call this theory $\mathsf{VNC}^1(\mathcal{L}_{\mathit{TreeRec}})$.

3 Constructing edge expanders

Here we define an iterative construction of a constant-degree edge expander family, and argue its edge expansion properties using simple combinatorial tools. The simplicity of the analysis will allow us (in Section 5) to formalize it within the system VNC^1. The construction is a variant of the iterative construction given by Reingold, Vadhan, and Wigderson [38], using the graph operations described next.

3.1 Graph operations

We define the graph operations that we will use to construct expanders.

Powering For a graph $G = (V, E)$ and an integer $k \geq 1$, the kth power G^k is the graph on vertices V where for each walk of length k from a vertex u to a vertex v in G there is an edge $u \sim v$ in G^k.

If Rot_G is the rotation map of G, then the rotation map of G^k is

$$Rot_{G^k}(v, (i_1, \ldots, i_k)) = (w, (j_k, \ldots, j_1)),$$

where w is the vertex reached from v in G by edges $i_1, \ldots, i_k$ using the rotation map Rot_G, and $(j_k, \ldots, j_1)$ describes the same sequence of edges in reverse order from w's point of view. For instance, $Rot_G(v, i_1) = (v', j_1)$ for some $v' \in V$, then $Rot_G(v', i_2) = (v'', j_2)$ for some v'', etc.

Tensor product For graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, their tensor product $G_1 \otimes G_2$ is the graph on vertices $V_1 \times V_2$, where for every pair of edges $u \sim u'$ in G_1 and $v \sim v'$ in G_2 there is an edge $(u, v) \sim (u', v')$ in $G_1 \otimes G_2$.

If Rot_{G_1} and Rot_{G_2} are the rotation maps of G_1 and G_2, respectively, then the rotation map of $G_1 \otimes G_2$ is

$$Rot_{G_1 \otimes G_2}((v, w), (i, j)) = ((v', w'), (i', j')),$$

where $Rot_{G_1}(v, i) = (v', i')$ and $Rot_{G_2}(w, j) = (w', j')$.

Replacement product For a D-regular graph $G = (V, E)$ on n vertices and a d-regular graph $H = (V', E')$ on D vertices, the replacement product $G \circ H$ is a $2d$-regular graph on nD vertices $\{(v, i) \mid v \in V,\ 1 \leq i \leq D\}$. The graph $G \circ H$ has the edges $\{(v, i) \sim (v, j) \mid v \in V,\ i \sim j \in E'\}$ as well as, for every edge $v \sim w$ in G such that w is the ith neighbor of v, and v is the jth neighbor of w (i.e., $Rot_G(v, i) = (w, j)$), $G \circ H$ has d parallel edges between (v, i) and (w, j).

If Rot_G and Rot_H are the rotation maps of G and H, respectively, then the rotation map of the $G \circ H$ is

$$Rot_{G \circ H}((v, i), j) = \begin{cases} ((v, i'), j') \text{ for } Rot_H(i, j) = (i', j') & \text{if } j \leq d \\ ((w, i'), j) \text{ for } Rot_G(v, i) = (w, i') & \text{if } j > d. \end{cases}$$

Adding self-loops For a d-regular graph $G = (V, E)$, the graph $\bigcirc G$ is the $2d$-regular graph obtained from G by adding d parallel self-loops around every vertex of G; note that we count every self-loop around vertex v as one edge $v \sim v$.

If Rot_G is the rotation map of G, then the rotation map of $\bigcirc G$ is

$$Rot_{\bigcirc G}(v, i) = \begin{cases} Rot_G(v, i) & \text{if } i \leq d \\ (v, i) & \text{if } i > d. \end{cases}$$

3.2 Effect of graph operations on edge expansion

For the operation of adding self-loops, the following lemma is obvious.

▶ **Lemma 3** (Self-loops). *If G is a d-regular graph with edge expansion ϵ, then the graph $\bigcirc G$ is $2d$-regular with edge expansion $\epsilon/2$.*

▶ **Lemma 4** (Powering). *Let G be a d-regular graph with edge expansion ϵ. For every integer $k \geq 1$, the powered graph $(\bigcirc G)^k$ has edge expansion at least*

$$\frac{1}{2} \cdot \left(1 - \left(1 - \frac{\epsilon^2}{4}\right)^{k/2}\right).$$

Proof Sketch. Our analysis is done in two stages. First, we use the result of Mihail [28] showing that a random k-step walk on an edge expander $\bigcirc G$ quickly converges to the uniform distribution over the vertices of $\bigcirc G$. Then we show that such convergence to the uniform distribution implies good edge expansion of $(\bigcirc G)^k$, using some ideas from [2].

Mihail [28] gave a combinatorial proof of the following result showing the exponentially fast convergence of a random walk on a regular graph to the uniform distribution.

▶ **Claim 5** ([28]). *Let G be a d-regular graph with edge expansion ϵ. Let A be the normalized adjacency matrix of $G' = \bigcirc G$. Let π be any initial distribution on vertices of G', and let u be the uniform distribution on vertices of G'. Then*

$$\|A^k\pi - u\|^2 \leq (1 - (\epsilon^2/4))^k \cdot \|\pi - u\|^2.$$

Let $G' = \bigcirc G$, and let $G'' = (G')^k$. Next we relate the edge expansion of G'' to the mixing time of a k-step random walk on G'. Let u denote the uniform distribution on the vertices of G''. For a subset U of vertices of G'', we denote by u_U the probability distribution that is uniform over U, i.e., every vertex in U gets weight $1/|U|$, and every vertex outside of U gets weight 0. We denote by χ_U the characteristic vector of the set U (whose ith entry is 1 if $i \in U$, and is 0 otherwise). The following claim can be proved along the lines of [2].

▶ **Claim 6.** *Suppose $G'' = (V, E)$ is a regular graph on n vertices, with normalized adjacency matrix A such that for some $\delta > 0$ the following holds: for every subset $U \subset V$ of size at most $|V|/2$,*

$$\|Au_U - u\|^2 \leq \delta \cdot \|u_U - u\|^2.$$

Then G'' has edge expansion at least $(1 - \sqrt{\delta})/2$.

Now, by Claim 5, we get for the normalized adjacency matrix A of the graph $\bigcirc G$ and for every subset $U \subset V$ that

$$\|A^k u_U - u\|^2 \leq (1 - (\epsilon^2/4))^k \cdot \|u_U - u\|^2.$$

Applying Claim 6 concludes the proof of Lemma 4. ◀

▶ **Lemma 7** (Tensoring). *Let $G = (V_G, E_G)$ be a d_G-regular graph with $d_G/2$ self-loops at every vertex and $H = (V_H, E_H)$ be a d_H-regular graph with $d_H/2$ self-loops at every vertex. If G has edge expansion ϵ_G and H has edge expansion ϵ_H, then the tensor product graph $G \otimes H$ has edge expansion at least $\min\{\epsilon_G, \epsilon_H\}/50$.*

Proof Sketch. First, we give some intuition. Suppose G is a d_G-regular graph on n_G vertices, and H is d_H-regular graph on n_H vertices. As a "warm-up", consider the special case of a subset of vertices S of the tensor product $G \otimes H$ such that $S = A \times B$. Moreover, assume that $|B| < n_H/2$. Then at least $\epsilon_H d_H |B|$ edges are leaving the set B in graph H. Each of these edges paired up with an edge from A will be an edge leaving $A \times B$ in $G \otimes H$, yielding a total of at least $\epsilon_H d_H |B| d_G |A|$ edges leaving $A \times B$. After normalization (division by $d_G d_H |A||B|$), this yields edge expansion ϵ_H from the set S. In the case, B is larger than $n_H/2$, but A is smaller than $n_G/2$, we can use the edge expansion of A, to obtain the edge expansion at least ϵ_G from S.

For general sets S of vertices in $G \otimes H$, we consider the characteristic matrix of S, which is an $n_G \times n_H$ 0-1 matrix with (i,j)th entry being 1 iff $(i,j) \in S$. We then argue that it is possible to remove some rows or some columns of this matrix so that the resulting matrix has a constant fraction of 1's of the original matrix (i.e., we removed only a constant fraction of vertices from S), and either every row or every column has at most some constant fraction of 1's.

Suppose we have the former case (the other case is treated similarly). That is, we removed some rows of the characteristic matrix of S to obtain a new subset S' that has the form $\{a_1\} \times B_1 \cup \cdots \cup \{a_k\} \times B_k$, where $a_i \in V_G$ and $B_i \subset V_H$, and moreover, each $|B_i|$ is at most some constant fraction of n_H. Then for each B_i, we can use edge expansion of H to argue that ϵ_H fraction of edges from B_i are leaving B_i. Ideally, we would like then to argue that each such edge, when paired up with any edge from vertex a_i, will leave S'. This may not be true, however, as such an edge may go to some vertex in $\{a_j\} \times B_j$. To circumvent this problem, we use the assumption that both of our graphs G and H have many self-loops around every vertex (say, half of the degree). In that case, it is easy to argue that each edge leaving B_i in H, when paired up with any self-loop around a_i, yields an edge of $G \otimes H$ that leaves S. Since the number of self-loops around a_i is at least half the degree of G, this yields edge expansion at least $\epsilon_H/2$ from each set $\{a_i\} \times B_i$. Since S' is the union of the pairwise disjoint such sets, we get the edge expansion at least $\epsilon_H/2$ from S'. Finally, since S' contains a constant fraction of vertices from S, we conclude that the edge expansion from S is at least $\Omega(\epsilon_H)$. ◀

▶ **Lemma 8** (Replacement [4]). *Let $G = (V_G, E_G)$ be a D-regular graph on n vertices, and let $H = (V_H, E_H)$ be a d-regular graph on D vertices. If G has edge expansion ϵ_G and H has edge expansion ϵ_H, then $G \circ H$ has edge expansion at least $\epsilon_G^2 \epsilon_H/48$.*

Proof Sketch. The proof idea is to partition a given subset S of vertices of $G \circ H$ into n clusters $(\{a_1\} \times B_1) \cup \cdots \cup (\{a_n\} \times B_n)$, where each $a_i \in V_G$ and $B_i \subseteq V_H$. View the clusters where $|B_i|$ is at most some fraction of $|V_H|$ as light, and the remaining clusters as heavy. For every light cluster, one can use the expansion of H to lower-bound the expansion of B_i (within the copy of H associated with vertex a_i of G). If there are many vertices in light clusters, we get a good lower bound on the edge expansion of S. Otherwise, there are many vertices in heavy clusters. Using the expansion properties of G, one can argue in this case that there will be many edges between the set of vertices in heavy clusters and the vertices outside S. ◀

3.3 Construction

With the analysis of graph operations in hand, we can now present our iterative construction of edge expanders that will be shown formalizable in VNC[1]. Let G_0 be a $(2d)$-regular graph of constant size, where d is some constant. Let ϵ_0 be the edge expansion of G_0 such that

$\epsilon_0 \geq 1/1296$. Such a graph G_0 exists (by a counting argument) and can be found in constant time, using exhaustive search. Given G_0, we will define a bigger graph G_1 that is also $(2d)$-regular and has edge expansion at least $1/1296$. In general, given a $(2d)$-regular graph G_i with edge expansion at least $1/1296$, we define G_{i+1} as follows:

$$G_{i+1} \;=\; ((\bigcirc((\bigcirc G_i) \otimes (\bigcirc G_i)))^c) \circ H, \tag{2}$$

where c is some constant to be specified later, and H is a d-regular expander graph on $(2(4d)^2)^c$ vertices, with edge expansion at least $1/3$. Again, such a graph can be found using exhaustive search.

▶ **Theorem 9.** *There is a constant c such that the graph $G_{i+1} = (V_{i+1}, E_{i+1})$ defined from $G_i = (V_i, E_i)$ as above has the following parameters:*

- $|V_{i+1}| = |V_i|^2 \cdot D$, *where* $D = (2(4d)^2)^c$,
- *the degree of G_{i+1} is $2d$,*
- *the edge expansion of G_{i+1} is at least $1/1296$.*

Proof. The bounds on the size and the degree of G_{i+1} follow easily from the definitions of the graph operations used to define G_{i+1} from G_i. Let $\epsilon \geq 1/1296$ be the edge expansion of G_i. First, by Lemma 3, the edge expansion of $\bigcirc G_i$ is at least $\epsilon/2$. By Lemma 7, the edge expansion of $G' = (\bigcirc G_i) \otimes (\bigcirc G_i)$ is at least $\epsilon' = \epsilon/100$. By Lemma 4, the kth power of the graph $\bigcirc G'$ has edge expansion at least

$$\frac{1}{2} \cdot \left(1 - \left(1 - \frac{\epsilon^2}{40000}\right)^{k/2}\right).$$

Choose k to be a sufficiently large constant c so that the edge expansion of the cth power of our graph, as given by the formula above, is at least $1/3$. Finally, by Lemma 8, the edge expansion of the graph G_{i+1} is at least $(1/3)^3/48 = 1/1296$. This completes the proof. ◀

We give also a modified construction of expanders that allows explicit constructions of edge expanders $\tilde{G}_i = (\tilde{V}_i, \tilde{E}_i)$ with $|\tilde{V}_i| = 2^i$, and more generally of edge expanders on exactly M vertices for arbitrary $M \geq 1$.

Let c be a constant. Choose the constant d to be a sufficiently large power of two, $d = 2^\ell$, so that there is a d-regular graph H on $(2(4d)^2)^c$ vertices with edge expansion at least $1/3$ and so that for all $i \leq c\ell + 7$, there are $2d$-regular graphs $\tilde{G}_i$ on 2^i vertices with edge expansion at least $1/1296$. These graphs H and $\tilde{G}_i$ can be found by exhaustive search. We construct expander graphs $\tilde{G}_i$ with edge expansion $\geq 1/1296$. For $i > 2c\ell + 7$, let $i' = \lfloor (i - 2c\ell - 5)/2 \rfloor$ and $i'' = \lceil (i - 2c\ell - 5)/2 \rceil$, so $i = i' + i'' + 2c\ell + 5$. Define

$$\tilde{G}_i \;=\; ((\bigcirc((\bigcirc \tilde{G}_{i'}) \otimes (\bigcirc \tilde{G}_{i''})))^c) \circ H. \tag{3}$$

▶ **Theorem 10.** *There is a constant c such that the graph $\tilde{G}_i = (\tilde{V}_i, \tilde{E}_i)$ defined as above has the following parameters:*

- $|\tilde{V}_i| = 2^i$,
- *the degree of $\tilde{G}_i$ is $2d$,*
- *the edge expansion of $\tilde{G}_i$ is at least $1/1296$.*

Now that we have constructed edge expanders of sizes 2^i, it is easy to obtain an edge expander $\tilde{G}$ of a given *arbitrary size* M. For this, choose i so that $2^{i-1} < M \leq 2^i$. Partition the vertices of $\tilde{G}_i$ into M disjoint subsets each of size 1 or 2. Define the graph $\tilde{G}$ by collapsing each of these subsets of vertices of $\tilde{G}_i$ into a single vertex of $\tilde{G}$, and inheriting the edges from

the all of the nodes in the subset. It is easy to see the degree of $\tilde{G}$ is at most $4d$; by adding extra self-loops, we get a new graph that is $4d$-regular. It is also easy to show that this new graph has expansion at least $\epsilon/2$ where $\epsilon = 1/1296$.

We get the following.

▶ **Theorem 11.** *Fix constants c and d as above. There is a family of $4d$-regular expanders $\tilde{G}$ on M nodes, for any $M \geq 1$, with edge expansion at least $1/2592$.*

Moreover, there is a deterministic polynomial-time algorithm that, given the name of a vertex v (in binary) of $\tilde{G}$ and an index $i \in [2d]$, outputs the value $Rot_{\tilde{G}}(v, i)$. Furthermore, there is an alternating linear time algorithm which accepts the graph of $\tilde{G}$; namely, it accepts exactly the triples of the form $\langle v, i, Rot_{\tilde{G}}(v, i)\rangle$.

It may be unexpected that we discuss alternating linear time, but the point is that this is what we need for the formalization of our arguments in the bounded arithmetic theory VNC^1 in Section 5. For that, the important thing is the computational complexity of Rot_{G_k} as a function of the size $|V_k|$ of the graph, whereas Theorem 11 expresses runtimes in terms of the size of the name of the vertex. But, the alternating linear time algorithm of Theorem 11 will be viewed as an alternating logarithmic time algorithm for purposes of formalization in VNC^1. (In the same setting, the polynomial time algorithm would be a polylogarithmic time algorithm, and it is open whether such algorithms can in general be formalized in VNC^1.)

4 Constructing bipartite vertex expanders

Jeřábek [24] needs the existence of certain bipartite vertex expanders to formalize the AKS sorting networks in VNC^1_*. We define these graphs next. Recall that, for a set S of nodes in a graph G, $\Gamma(S)$ denotes the set of all neighbors of vertices in S.

Given constants $\alpha \in (0, 1)$ and $A > 1$, a *bipartite (α, A) vertex expander* is a bipartite graph $G = (L \cup R, E)$, where $|L| = |R| = m$, such that

1. the degree of G is at most A, and
2. for all $\ell \leq m$, every set $S \subseteq [m]$ of vertices in either partition with $|S| \geq \alpha\ell$ has $|\Gamma(S)| \geq (1 - \alpha)\ell$.

That is, for every set of vertices of size at least $\alpha\ell$ in one partition, there are at least $(1-\alpha)\ell$ neighbors in the other partition.

The assumption required by [24] is:

> For $\alpha = 1/600$, there exist a constant A and a parameter-free NC^1_* function $G(m)$ such that VNC^1_* proves "$\forall m \in \mathbb{N}$, $G(m)$ is an (α, A) bipartite vertex expander on $m + m$ vertices".

We will argue that such bipartite vertex expanders can be efficiently obtained from our edge expanders defined above.

▶ **Theorem 12.** *For any constant $0 < \alpha < 1$, there exist a constant $A \geq 1$ and an efficient (uniform NC^1) algorithm that, for every $m \in \mathbb{N}$, computes the rotation map of an (α, A) bipartite vertex expander on $m + m$ vertices.*

Proof. We use the edge expander $\tilde{G}$ constructed in Theorem 11 with $M = m$. The graph $\tilde{G} = (\tilde{V}, \tilde{E})$ has $|V| = m$, degree $4d$, and expansion at least $\epsilon/2$, where $\epsilon = 1/1296$. Starting with $\tilde{G}$, we will

1. Convert the edge expander $\tilde{G}$ into a vertex expander, and
2. Turn the latter vertex expander into a bipartite (α, A) vertex expander on $m+m$ vertices.

1. Getting a vertex expander from an edge expander: Let $G = (V, E)$ be the graph $\tilde{G}$ on m nodes constructed above, but with a self-loop added to every node. So G has constant degree $4d + 1$.

We have for every set $S \subseteq V$ of size $|S| \leq m/2$ that at least $\epsilon(2d)|S|$ edges are leaving S in $\tilde{G}$. As the degree of $\tilde{G}$ is $4d$, we conclude that the neighborhood $\Gamma(S)$ of S in G contains at least $\epsilon \cdot (2d) \cdot |S|/(4d) = \epsilon' \cdot |S|$ distinct nodes from $\overline{S}$, where $\epsilon' = \epsilon/2$. As G has self-loops around every node, we get

$$|\Gamma(S)| \geq (1 + \epsilon') \cdot |S|, \tag{4}$$

for every subset S of G with $|S| \leq m/2$.

Consider the power graph G^i, for any $i \geq 1$. Applying Eq. (4) inductively, we get for every subset S of G^i with $|S| \leq m/2$, and for every $i \geq 1$ that

$$|\Gamma_{G^i}(S)| \geq \min\{m/2, (1 + \epsilon')^i \cdot |S|\}. \tag{5}$$

Now let S be a subset of V of size $|S| \geq m/2$. We have $|\Gamma^+(S)| \geq \epsilon' \cdot |\overline{S}|$, where $\Gamma^+(S) = \Gamma(S) \cap \overline{S}$ is the set of new neighbors of S. It follows that

$$|\overline{\Gamma(S)}| \leq (1 - \epsilon') \cdot |\overline{S}|. \tag{6}$$

Applying Eq. (6) inductively, we get for every $i \geq 1$, and for every subset S of V of size $|S| \geq m/2$ that

$$|\overline{\Gamma_{G^i}(S)}| \leq (1 - \epsilon')^i \cdot |\overline{S}|. \tag{7}$$

▶ **Claim 13.** *There exists a constant $t' = t'(\alpha, \epsilon')$ such that, for every $\ell \leq m$ and every set S of $G^{t'}$ with $|S| \geq \alpha\ell$, we have $|\Gamma_{G^{t'}}(S)| \geq (1 - \alpha)\ell$.*

Proof of Claim 13. Consider two cases: $\ell \leq m/2$, and $\ell > m/2$. If $\ell \leq m/2$, then by Eq. (5) we get for $t_1 = \lceil \log_{1+\epsilon'}(1/\alpha) \rceil$ that

$$|\Gamma_{G^{t_1}}(S)| \;\geq\; \min\{m/2, (1 + \epsilon')^{t_1} \cdot \alpha\ell\} \;\geq\; \min\{m/2, \ell\} \;=\; \ell.$$

If $\ell > m/2$, then $|\overline{S}| \leq m - \alpha\ell < (1 - (\alpha/2)) \cdot m < m$. For $t_2 = \lceil (\log 1/\alpha)/(\log 1/(1 - \epsilon')) \rceil$, we get

$$|\overline{\Gamma_{G^{t_2}}(S)}| \;\leq\; (1 - \epsilon')^{t_2} \cdot m \;\leq\; \alpha \cdot m,$$

and hence, $|\Gamma_{G^{t_2}}(S)| \geq (1-\alpha)m \geq (1-\alpha)\ell$. Taking $t' = \max\{t_1, t_2\}$ concludes the proof. ◀

2. Getting a bipartite vertex expander: Let $G^{t'}$ be the vertex expander defined above. Observe that it has m nodes, and has constant degree $A = (4d + 1)^{t'}$. We turn this graph into a bipartite graph by taking two copies of the vertices of $G^{t'}$, denoted by L and R, connecting nodes $i \in L$ and $j \in R$ by an edge iff $\{i, j\}$ is an edge of $G^{t'}$. Claim 13 implies that the resulting graph is an (α, A) vertex expander.

Finally, the explicitness of this construction of (α, A) vertex expanders can be argued similarly to the case of the edge expanders of Theorem 11: we trace the construction of $G^{t'}$ to get an efficient (uniform NC^1) algorithm for computing the rotation map of the corresponding bipartite (α, A) expander on $m + m$ vertices. ◀

5 Formalizing the construction in bounded arithmetic

This section discusses the formalization of the expander graph construction in the theory VNC^1 of bounded arithmetic. A high-level description of how we formalize the expander graph construction in VNC^1 is as follows:

1. The first step is to establish (in Section 5.4) that VNC^1 can define the operations of graph powering, replacement product, and tensoring. From this it follows that VNC^1 can carry out the definition of G_{i+1} from G_i, for the graphs G_i defined in Section 3. Similarly, VNC^1 can carry out the construction of $\tilde{G}_i$ from $\tilde{G}_{i'}$ and $\tilde{G}_{i''}$ as in (3).
2. For the second step, we wish to use induction on t to prove the existence of the graph G_t for suitable t. However, since VNC^1 does not support induction on Σ^B_1-formulas, we cannot use the usual induction axioms for VNC^1. Instead, we exploit the fact that the graph G_{i+1} has size quadratic in the size of G_i, namely $|G_{i+1}| = \Theta(|G_i|^2)$. This large growth rate allows us to use Σ^B_1-induction to prove the existence of G_t for arbitrary (first-order) integers t. For this, Theorems 16 and 17 of Section 5.3 prove that the needed induction principle is provable in VNC^1. The intuition is that the computational content of the induction axioms corresponds to composing logarithmic depth circuits, and that since the G_i's are growing quadratically, arbitrary composition of logarithmic depth circuits for the G_i's yields a circuit which is still of only logarithmic depth.
   The same Σ^B_1-induction will also be used to prove the existence of the graphs $\tilde{G}_i$, exploiting the fact that the size of $\tilde{G}_i$ is quadratic in the sizes of $\tilde{G}_{i'}$ and $\tilde{G}_{i''}$.
3. Theorems 16 and 17 give the needed induction principle for handling compositions of circuits, but more work is needed for VNC^1 to formalize the iterated composition of circuits. What we mean by "iterated composition" of circuits is that there are multiple circuits (about $|(|t|)|$ many circuits) which are arranged with the outputs of one circuit feeding into the inputs of the next circuit. To formalize this circuit composition in VNC^1, we need to modify Cook and Morioka's definition [16] of *TreeRec* tree recursion in VNC^1. The problem with the *TreeRec* form of tree recursion is that the second order inputs to a circuit defined by tree recursion are not used at the input gates of the circuit, but rather are used throughout the circuit, indeed potentially at every gate in the circuit. To fix this, Section 5.2 introduces a modified version of tree recursion, called *TreeRec*$'$, which allows the use of second order inputs $X_0(i)$ only as input values. This allows composition of circuits using the inputs X_0 for the iteratively computed values. The *TreeRec*$'$ tree recursion and the new induction principle of Section 5.3 then suffice to define G_t by using recursively the definition of G_{i+1} from G_i.
4. The fourth step is to prove the expansion properties of G_{i+1} follow from those of G_i. Or, more precisely, proving that if G_{i+1} does not have the desired edge expansion then G_i also does not.
5. The fifth step is to use induction on t to prove the expansion properties for G_t. This is done in Theorem 21; its proof again utilizes the induction principle introduced in Section 5.3. This shows that VNC^1 can prove the existence of expander graphs.
6. The sixth, and final step, is to note that the proof of Theorem 12 can be carried out in VNC^1, so VNC^1 proves the existence of bipartite vertex expanders.

This proof is given below. We start by proving some useful properties of VNC^1 in Sections 5.1–5.3. We show in Section 5.4 that VNC^1 can express relevant graph properties. Section 5.5 shows that the edge expansion properties of our graph operations can be proved within VNC^1.

5.1 Defining NC^1 functions within VNC^1

Cook and Morioka [16, Lemma 13] show that $\mathsf{VNC}^1(\mathit{TreeRec})$ can prove the $\Sigma^B_0(\mathit{TreeRec})$-COMP axioms. They then define the FNC^1 functions F by using $\Sigma^B_0(\mathit{TreeRec})$-formulas $\varphi(i,\vec{x},\vec{X})$ and terms $t(\vec{x},\vec{X})$ and defining the string $F(\vec{x},\vec{X})$ by[2]

$$F(\vec{x},\vec{X})(j) \;\leftrightarrow\; j{<}t(\vec{x},\vec{X}) \wedge \varphi(j,\vec{x},\vec{X}). \tag{8}$$

They also show that the Σ^B_1-definable functions of VNC^1 are precisely the FNC^1 functions [16, Theorem 17]. Recall that a Σ^B_1-definition is given by VNC^1 proof of $(\exists!Y)\varphi(\vec{x},\vec{X},Y)$ where $\varphi \in \Sigma^B_1$; this serves as a definition of the string function $\vec{x},\vec{X} \mapsto Y$.

The definition of FNC^1 functions using (8) is equivalent to the usual definition of the FNC^1 functions as the functions whose bit graphs are computable in U_{E^*}-uniform NC^1, or equivalently are computable in $\mathsf{ALogTime}$. Those functions are computed by a family $\{C_n\}_n$ of fanin ≤ 2 Boolean circuits, taking inputs of length n and having depth $O(\log n)$. The U_{E^*}-uniformity condition was defined by Ruzzo [40] and means that the circuits C_n are described by two functions $g(i,n)$ and $p(i,w,n)$ which are computable in the linear time hierarchy (equivalently, they have Σ^B_0 graphs). The first function $g(i,n)$ returns the type of gate i in C_n. The second function $p(i,w,n)$ takes as input also a $w \in \{0,1\}^*$: the bits of w describe a path in the circuit starting at gate i and following successively the first or second input to gates according to the bits of w. The value of $p(i,w,n)$ is the index of the gate reached by following this path specified by w starting from gate i in C_n. The functions g and p are in the linear time hierarchy; however, since they have inputs of length $O(\log n)$, they run in time $O(\log n)$ using a constant number of alternations. For more details, see [40].

We will need to carefully analyze the effect of composing FNC^1 functions; for this reason it is important that the existence of U^*_E-uniform NC^1 circuits for FNC^1 functions can be proved by the theory VNC^1. This follows from Theorem 15 below.

5.2 A modified tree recursion

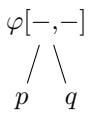
TreeRec acts like a fanin two, Boolean circuit where the internal gate types are given by $\varphi = \varphi(i,\vec{x},\vec{X})$. A disadvantage of this definition of *TreeRec* is that the side parameters $\vec{X}$ can be used unrestrictedly by the Σ^B_0-formulas φ and ψ. The formula $\varphi(i,\vec{x},\vec{X})$ defines the type of gate number i when the circuit has $\vec{x},\vec{X}$ as inputs. Likewise, $\psi(i,\vec{x},\vec{X})$ defines the True/False value of the i-th input. This differs from the usual conventions of having a circuit have fixed gate types, and having the inputs affect only the values of input gates. It also makes it difficult to define the notion of composing circuits, with the outputs of one family of circuits serving as the inputs to another circuit.

We define a new formulation of tree recursion called *TreeRec′* to address this problem. In a *TreeRec′* definition, one of the second order inputs, X_0, will serve as an "ordinary" input to the circuit, with the values $X_0(j)$ specifying the True/False values on inputs to the circuit. The other second order inputs, $\vec{X}'$, can be used to define gate types similarly as is done by *TreeRec*. This allows recursive computations on the value X_0 to be formalized with composition of circuits.

We assume X_0 is one of the side string parameters $\vec{X}$, so $\vec{X}$ is $X_0,\vec{X}'$. We modify the definition of *TreeRec* so that the values $X_0(i)$ are used only as inputs to the *TreeRec* circuit,

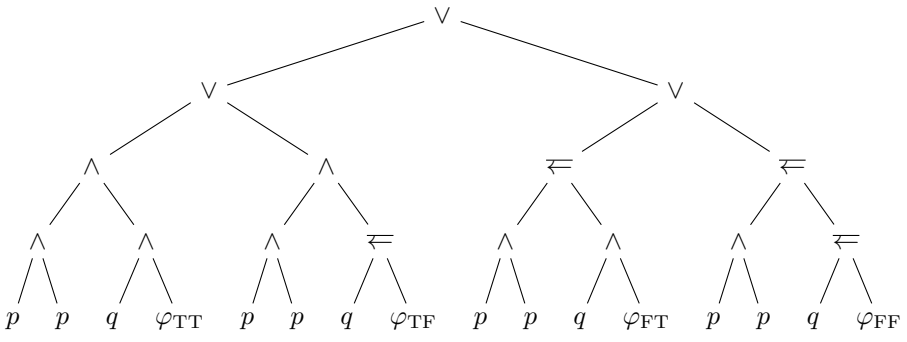
[2] This definition of FNC^1 is same as what Cook and Morioka [16] call "the function symbols in $\mathsf{VNC}^1(\mathsf{FNC}^1)$". We use just "$\mathsf{FNC}^1$" to keep the notation less cumbersome.

and are not used to determine the gate types; in particular, X_0 is not used by φ. The basic construction for the definition of *TreeRec*′ is that a single gate in a *TreeRec* circuit, of gate type $\varphi[-,-]$:



(9)

is replaced by a small tree of binary gates



(10)

Here the binary gate $r \Leftarrow s$ is $\neg r \wedge s$; and the values φ_{pq} are the truth values of $\varphi(i, a, \vec{x}, \vec{X})[p, q]$. By inspection, the circuit is depth four and fanin two: the top $\vee$ gate branches on the value of p; the next two $\vee$ gates branch on q. The last two levels select the correct value of φ_{pq}, for $p =$ T,F and $q =$ T,F based on the values of p and q. In other words, the circuit (10) implements a "lookup table", using the values p and q to select the appropriate value φ_{pq}. Assuming that the four values of φ_{pq} are correctly computed, the effect of replacing the binary gates (9) with the circuits (10) gives a circuit of depth $4|a|$ computing the same result as the original *TreeRec* circuit of depth $|a|$.

We wish to replace the four leaf nodes of (10) labelled φ_{pq} with Boolean circuits which have as inputs only the values $X_0(i)$. Since φ is Σ^B_0-formula, such circuits can easily be described by a polynomial time function of $i, \vec{x}, \vec{X}'$. These circuits are formed by applying the Paris-Wilkie transformation to φ, namely by replacing bounded quantifiers in φ with conjunctions and disjunctions, and hardcoding the values of $\vec{x}$ and $\vec{X}'$ (but not X_0) as constants. The result is that each leaf φ_{pq} of the circuit (10) can be replaced by a fanin two circuit which (a) has as inputs only $X_0(j)$'s and constants, (b) is size $\leq q(|a|, |\vec{x}|)$ and depth $\leq |q(|a|, |\vec{x}|)|$ for some polynomial q, and (c) there is a Σ^B_0-definable number function $f(p, q, i, a, \vec{x}, \vec{X})$ of VNC^1 which outputs a succinct description of the circuit. VNC^1 is able to straightforwardly define f and prove all these properties.

With this construction in hand, we define a modified version of tree recursion:

▶ **Definition 14.** Let $\varphi(i, a, \vec{x}, y_0, \vec{X}')[p, q]$ be a Σ^B_0-formula and let $k(i, a, \vec{x}, y_0, \vec{y}')$ be a Σ^B_0-definable number function. The Σ^B_0-*TreeRec*′ property for φ and k is given by:

$$B^{\varphi,k}(a, \vec{x}, X_0, \vec{X}', Z) = (\forall i{<}a)[\,(Z(i) \leftrightarrow \varphi(i, a, \vec{x}, |X_0|, \vec{X}')[Z(2i{+}1), Z(2i{+}2)]) \\ \wedge\ (Z(a+i) \leftrightarrow X_0(k(i, a, \vec{x}, |X_0|, |\vec{X}'|)))\,].$$

The *defining axioms for the predicate symbols* $R^{\varphi,k}(i, a, \vec{x}, X_0, \vec{X}')$ are the formulas

$$B^{\varphi,k}(a, \vec{x}, X_0, \vec{X}', R^{\varphi,k}) \qquad \text{and} \qquad i{\geq}2a \rightarrow \neg R^{\varphi,k}(i, a, \vec{x}, X_0, \vec{X}'). \tag{11}$$

Note that the gate type depends only on $|X_0|$, not on the values of $X_0(\cdot)$. VNC^1 proves that (11) uniquely specifies all values of $R^{\varphi,k}$. Furthermore, it is not hard to see that VNC^1 proves the existence of string objects satisfying the conditions of (11). Thus, we may conservatively extend $\mathsf{VNC}^1(\mathit{TreeRec})$ by adding all these predicate symbols along with their defining axioms. The resulting theory is called $\mathsf{VNC}^1(\mathit{TreeRec}, \mathit{TreeRec}')$.

The main advantage of $\mathit{TreeRec}'$ definitions is that they can explicitly give U_{E^*}-uniform log-depth circuits. For this, we assume that X_0 is the only second-order input (so $\vec{X}'$ is missing). We also assume that $a = s(\vec{x}, |X_0|)$ for some V^0-term s. It is usually convenient to assume in addition that each $x_i < |X_0|^{O(1)}$, so that we can think of $|X_0|$ as the size of the input (up to a polynomial); in fact, often $\vec{x}$ is missing, so the only first-order input is $|X_0|$. The other condition needed for U_{E^*}-uniformity is that there must be a linear time hierarchy algorithm (i.e., a Σ^B_0 formula) determining the extended connection language for the connectivity of gates in the circuit. Since the circuit is formed as a binary tree, with a natural numbering system for gates, the extended connection language of the circuit is trivial. Specifically, suppose $w \in \{0,1\}$ is a string of bits and i is a gate. Interpret bits "0" and "1" as selecting the first or second input to a gate, and let w specify a path starting at gate i, and traversing inputs according to the bits of w. The gate at the end of this path is gate i' where i' has binary representation obtained by concatenating the binary representation of i and the string w. The type of gate i can be defined with a Σ^B_0-formula using the Σ^B_0-formula φ and the Σ^B_0-defined function k. Thus, with the assumptions stated above, a $\mathit{TreeRec}'$ definition defines a U_{E^*}-uniform circuit.

The next theorem states that every $\Sigma^B_0(\mathit{TreeRec})$-property has log-depth, fanin two, Boolean circuits in the form used by $\mathit{TreeRec}'$.

▶ **Theorem 15.** *Let $\chi(\vec{x}, X_0, \vec{X}')$ be a $\Sigma^B_0(\mathit{TreeRec})$-formula. Then there are a Σ^B_0-formula $\varphi(i, a, \vec{x}, y_0, \vec{X}')$, a Σ^B_0-defined function $k(i, a, \vec{x}, y_0, |\vec{y}'|)$, and a V^0-term $s(\vec{x}, |X_0|, |\vec{X}'|)$ so that $\mathsf{VNC}^1(\mathit{TreeRec}, \mathit{TreeRec}')$ proves*

$$\chi(\vec{x}, X_0, \vec{X}') \;\leftrightarrow\; R^{\varphi,k}(0, s(\vec{x}, |X_0|, |\vec{X}'|), \vec{x}, X_0, \vec{X}').$$

$\Sigma^B_0(\mathit{TreeRec})$-properties may involve composing multiple *TreeRec* predicates with built-in function symbols, then combining them with Boolean operations and first-order quantifiers. Theorem 15 states that any such property χ may expressed as a $\mathit{TreeRec}'$: the advantage is that this gives an explicit NC^1 representation of χ; namely in terms of logarithmic depth Boolean circuits. "Logarithmic" means as a function of the values $\vec{x}$ and of the sizes $|X_0|, |\vec{X}'|$ of the second order inputs X_0, X'.

5.3 A conservation result

We prove the closure of VNC^1 under a rule of inference based on a "telescoping" iteration. This turns out to be exactly what is needed for the formalization of the expander graph construction inside VNC^1. We write $\sqrt{a}$ for the greatest integer at most $\sqrt{a}$.

▶ **Theorem 16.** *Suppose $\chi(X)$ is a Σ^B_0-formula containing only X free, and let $\psi(a)$ be $(\exists X{\leq}a)\chi(X)$. Also suppose VNC^1 proves*

$$(\forall a)(\psi(a) \to \psi(\sqrt{a})). \tag{12}$$

Then VNC^1 proves $\psi(a) \to \psi(1)$, and thus also proves $\chi(Y) \to (\exists X{\leq}\,1)\chi(X))$.

Theorem 16 used a descending induction; a similar theorem holds also for ascending induction:

▶ **Theorem 17.** *Suppose $\varphi(X)$ is a Σ_0^B-formula containing only X free. Also suppose* VNC^1 *proves*

$$\varphi(Y) \to (\exists X)(|X| \geq |Y|^2 \wedge \varphi(Y)).$$

Then VNC^1 *proves* $(\exists Y)\varphi(Y) \to (\forall x)(\exists X)(|X| > x \wedge \varphi(X))$.

5.4 Expressing expander graph properties in VNC^1

We now discuss how VNC^1 can express properties about graphs, adjacency matrices, expansion properties, and graph constructions such as powering, tensor product and replacement product. A graph G on n vertices will be encoded in VNC^1 as a string object (a second order object). Here n is a number (a first-order object), and the intent is to represent G in terms of its adjacency matrix. The (i,j)-th entry of the adjacency matrix is the number of edges between vertices i and j. It is represented by a three-place second order predicate $A(i,j,k)$ where $A(i,j,k)$ is true when there are exactly k edges between i and j. (Strictly speaking, we should write $A(\langle i,j,k\rangle)$, but we suppress this notation.) Each i,j,k is a number (a first order object); it will be important that we always have $k < p(n)$ for some fixed polynomial p, since then k is Σ_0^B-definable from A,i,j, and we can write $k = A(i,j)$ for the value of k.

Row vectors and column vectors (containing numbers) are likewise representable by strings, with $A(i,k)$ meaning that the i-th entry of the vector is equal to k.

With these conventions it is easy for VNC^1 to Σ_0^B- or Σ_1^B-define many properties of the graph G encoded as above. We illustrate this with several examples.

- For $u < n$, the set of edges containing vertex u can be defined as the set

  $$E(\{u\}) \;=\; \{\langle u,v,k\rangle : (\exists k' \leq p(n))(k < k' \wedge A(u,v,k'))\}.$$

  Note this allows for multiedges. The degree of v is $|E(\{v\})|$ and can be Σ_0^B-defined with the *Numones* function. G has degree d if each $u \in [n]$ has degree d. There will always be a polynomial upper bound $p(n)$ on the degree.
- For $U \subset [n]$, the set $E(U,\overline{U})$ is defined similarly as

  $$E(U,\overline{U}) \;=\; \{\langle i,j,k\rangle : i \in U \wedge j \notin U \wedge (\exists k' \leq p(n))(k < k' \wedge A(i,j,k'))\}.$$

- Rational numbers p/q are represented by pairs of integers (p,q) (not necessarily in reduced form). The usual ordering $p/q < p'/q'$ is of course definable by $pq' < p'q$, where $q, q' > 0$. Pairs of rational numbers may be added or multiplied or divided as usual.
  The proof of the Cauchy-Schwarz theorem, and more generally our proofs of expansion properties, argue about sums of vectors of rational numbers. VNC^1 can define summations of vectors of integers [14], but it is not clear whether it can define summations of vectors of arbitrary rational numbers. This will be handled in our VNC^1 proofs by clearing the denominators so that we can argue about summations of integers instead of about summations of rational numbers. In our applications, the least common multiple of the denominators will be easily computed, making it easy to clear the denominators.
- The edge expansion of a degree d graph G can thus be defined by as in equation (1) with $V = [n]$. This, however, is not a Σ_1^B-definition, since it requires minimizing over all subsets $U \subset [n]$. Instead we can define the property "G has edge expansion $> p/q$" as

  $$(\forall U{<}n)\Big(0 < |U| \leq \frac{n}{2} \to \frac{|E(U,\overline{U})|}{d \cdot |U|} > \frac{p}{q}\Big).$$

  This is a Π_1^B-condition. Recall that "$(\forall U{<}n)$" is quantifying over all subsets of $[n]$.

- A rotation map is encoded by a second order object $Rot(u, i, v, j)$ with the meaning that the i-th edge of u is the same as the j-th edge of v. We can relate the rotation map Rot and the adjacency matrix A by letting the i-th edge from u to v be the edge $\langle u, v, k\rangle$ such that

$$|\{\langle u, i', v, j\rangle : Rot(u, i', v, j) \wedge i' < i\}| \;=\; k$$

  Furthermore, the adjacency matrix A is Σ^B_1-definable in terms of Rot, since $A(u, v) = k$ holds exactly when there are exactly k values $\langle i, j\rangle$ such that $Rot(u, i, v, j)$. Since v, j are uniquely determined by u, i, we also use the notation $Rot(u, i) = (v, j)$.
  It is also possible to Σ^B_1-define a canonical rotation map as a function of the adjacency matrix.

Graph operations are also readily defined by VNC^1:

- To add self-loops to convert a d-regular G to a $2d$-regular G', define the adjacency matrix $A'(u, v, k)$ as

$$(u \neq v \wedge A(u, v, k)) \vee (u = v \wedge (\exists k' \leq d)(A(u, v, k') \wedge (k = k' + d))).$$

- (Graph Powering.) Let $k > 1$ be fixed. VNC^1 can Σ^B_1-define the graph power G^k from G as follows. We write $\langle i_1, \ldots, i_k\rangle$ for an efficient sequence coding so that each $\langle i_1, \ldots, i_k\rangle$ is represented by an integer $< d^k$. Then $Rot(u, \langle i_1, \ldots, i_k\rangle) = (v, \langle j_1, \ldots, j_k\rangle)$ holds iff

$$(\exists \langle u_0, \ldots, u_k\rangle)[u_0{=}u \wedge u_k{=}v \wedge \bigwedge\nolimits_{s=1}^{k} (Rot(u_{s-1}, i_s) = (u_s, j_{k-s+1}))].$$

  Since k is fixed and each $u_i < n$, the quantifier is a bounded number quantifier.
- Similar arguments give Σ^B_1-definitions of Tensor Product and Replacement Product. The constructions are straightforward and we leave the details to the reader.

These constructions, along with Theorem 17, allow VNC^1 to prove the existence of the graphs G_i as defined by (2). Fix constants d and c, and fix a $(2d)$-regular G_0 with edge expansion ϵ_0. Also, fix a rotation map $Rot_0 = Rot_{G_0}$ for G_0. Given G_i and Rot_i, for $i \geq 0$, VNC^1 can prove the existence of G_{i+1} satisfying (2) along with the existence of Rot_{i+1}. Furthermore, by Theorem 17, VNC^1 can prove the existence of a second-order object encoding a sequence of graphs and rotation maps

$$(G_0, Rot_0),\ (G_1, Rot_1),\ (G_2, Rot_2),\ \ldots,\ (G_{|a|}, Rot_{|a|}), \tag{13}$$

so each G_{i+1} and associated rotation map Rot_{i+1} is obtained from G_i and Rot_i by Equation (2). Letting the constant $D = 2(4d)^2)^c$ as before, each G_i has $(|V_0| \cdot 4D)^{2^i}/D$ many vertices, provably in VNC^1. (See Theorem 9.) The size of G_{i+1} is greater than the square of the size of G_i; indeed, $|V_{i+1}| = D \cdot |V_i|^2$. Therefore, Theorem 17 applies, to show that VNC^1 can Σ^B_1-define the sequence (13) as function of a, and hence can Σ^B_1-define $G_{|a|}$ and $Rot_{|a|}$ as functions of a.

Similar, only slightly more complicated, arguments allow VNC^1 to prove the existence of the graphs $\tilde{G}_i$ as defined by (3). Now i can be an arbitrary first-order (integer) value $i = a$, not just a length $|a|$. Fix appropriate constants $d = 2^\ell$ and c, and for $i \leq 2c\ell{+}8$, fix graphs $\tilde{G}_i$ with edge expansion $\geq 1/1296$ and their rotation maps Rot_i. Using induction on Σ^B_0-formulas, VNC^1 proves the existence of a sequence of values $k_0, \ldots, k_s$ such that $k_0 = a$ and each $k_{i+1} = \lfloor (k_i{-}2c\ell{-}5)/2 \rfloor$, and such that s is the first value where $k_s < 2c\ell{+}7$. Given both $\tilde{G}_{k_{i+1}}$ and $\tilde{G}_{k_{i+1}+1}$ and their rotation maps $Rot_{k_{i+1}}$ and $Rot_{k_{i+1}+1}$, and using

the definition (3), VNC^1 can prove the existence of both $\tilde{G}_{k_i}$ and $\tilde{G}_{k_i+1}$ and their rotation maps. Furthermore, the sizes of $\tilde{G}_{k_i}$ and $\tilde{G}_{k_i+1}$ are both greater than the square of the size of $\tilde{G}_{k_{i+1}+1}$. Therefore, by Theorem 17 again, VNC^1 can prove the existence of a second-order object encoding a sequence of pairs of graphs and rotation maps:

$$(\tilde{G}_{k_s}, Rot_{k_s}, \tilde{G}_{k_s+1}, Rot_{k_s+1}), (\tilde{G}_{k_{s-1}}, Rot_{k_{s-1}}, \tilde{G}_{k_{s-1}+1}, Rot_{k_{s-1}+1}), \ldots \\ (\tilde{G}_{k_0}, Rot_{k_0}, \tilde{G}_{k_0+1}, Rot_{k_0+1}), \tag{14}$$

with successive pairs of expander graphs obtained via (3). Since $k_0 = a$, this shows that VNC^1 can Σ^B_1-define $\tilde{G}_a$ and Rot_a as functions of a.

It is immediate from the definition of G_i, using induction on i, that VNC^1 proves that each G_i has degree $2d$ (for the appropriate value of d). Likewise VNC^1 proves that each $\tilde{G}_i$ has degree $2d$. It is more difficult to prove that VNC^1 proves G_i and $\tilde{G}_i$ have the edge expansion properties of Theorems 9 and 10. This is discussed in the next sections.

5.5 Formalizing edge expansion properties in VNC^1

We prove that the graph operations can be analyzed in VNC^1. For $\emptyset \neq U \subsetneq V$, we denote by $\textit{edge-exp}_G(U)$ the *edge expansion* ratio defined as follows:

$$\textit{edge-exp}_G(U) \;=\; \frac{|E(U, \overline{U})|}{d \cdot \min\{|U|, |\overline{U}|\}}.$$

▶ **Lemma 18.** *Let k be even.* VNC^1 *proves the following: Suppose G^k is the graph power of G as defined in Section 5.4, and V is the common vertex set of G and G^k. Then*

$$(\exists U)[U \subset V \wedge \textit{edge-exp}_{(\bigcirc G^k)}(U) < [\tfrac{1}{2}\big(1 - \big(1 - \tfrac{\epsilon^2}{4}\big)^{k/2}\big)] \rightarrow (\exists U)[U \subset V \wedge \textit{edge-exp}_G(U) < \epsilon].$$

▶ **Lemma 19.** VNC^1 *proves the following: Let $G = (V_G, E_G)$ be a d_G-regular graph with $d_G/2$ self-loops at every vertex and $H = (V_H, E_H)$ be a d_H-regular graph with $d_H/2$ self-loops at every vertex. Let $\epsilon = \min\{\epsilon_G, \epsilon_H\}$. Then,*

$$(\exists U)[U \subset (V_G \otimes V_H) \wedge \textit{edge-exp}_{G \otimes H}(U) < \epsilon/50] \\ \rightarrow\; (\exists U)[U \subset V_G \wedge \textit{edge-exp}_G(U) < \epsilon_G] \vee (\exists U)[U \subset V_H \wedge \textit{edge-exp}_H(U) < \epsilon_H].$$

▶ **Lemma 20.** VNC^1 *proves the following: Let $G = (V_G, E_G)$ be a D-regular graph on n vertices, and let $H = (V_H, E_H)$ be a d-regular graph on D vertices. Let $\epsilon = \epsilon_G^2 \epsilon_H/48$, and let $V_{G \circ H}$ denote the vertices of $G \circ H$. Then,*

$$(\exists U)[U \subset V_{G \circ H} \wedge \textit{edge-exp}_{G \circ H}(U) < \epsilon] \\ \rightarrow\; (\exists U)[U \subset V_G \wedge \textit{edge-exp}_G(U) < \epsilon_G] \vee (\exists U)[U \subset V_H \wedge \textit{edge-exp}_H(U) < \epsilon_H].$$

Finally, the arguments in Section 3.3 also formalize in VNC^1 to combine Lemmas 18-20 to prove the existence of expander graphs. For this, we need to formulate the arguments so as to apply Theorem 16. We first how how to prove the existence of the edge expanders G_i in VNC^1. To talk about the edge expansion of G_i, we encode a subset U of V_i using a string Y of length exactly $|V_i| + 1 = (|V_0| \cdot 4D)^{2^i}/D + 1$, by letting $Y = U \cup \{|V_i|\}$. It follows from the discussion at the end of Section 5.4 that VNC^1 can Σ^B_1-define G_i as a function of $|V_i|$, hence as a function of Y.

Let $A(Y)$ express the conditions that (a) $|Y| = |V_i| + 1$ for some i, and (b) Y encodes a subset U of V_i such that $\textit{edge-exp}_{G_i}(U) < 1/1296$. The (contrapositive of the) argument in Section 3.3, formalized in VNC^1, shows that the following is VNC^1 provable:

$$(\exists Y \leq a) A(Y) \rightarrow (\exists Y \leq \sqrt{a}) A(Y). \tag{15}$$

For $i = 0$, this uses the fact that G_0 has edge expansion $\geq 1/1296$, and since G_0 is a constant graph, this can be checked by a enumerating all of the finitely many subsets.

Applying Theorem 16 to (15) gives that VNC^1 proves

$$(\exists Y{\leq}a)A(Y) \to (\exists Y{\leq}1)A(Y).$$

There are only four possible Y's with $|Y| \leq 1$. The righthand side, $(\exists Y{\leq}1)A(Y)$, is a false Σ^B_0-formula asserting a finite property. Hence, VNC^1 can trivially disprove $(\exists Y{\leq}1)A(Y)$ by direct evaluation. Therefore, VNC^1 proves $\neg(\exists Y)A(Y)$, i.e., can prove that any V_i must be an expander. This completes the proof of the following.

▶ **Theorem 21.** *There is a constant d so that VNC^1 proves the existence of arbitrarily large, degree $2d$ graphs with edge expansion $\geq 1/1296$. Namely, VNC^1 proves*

$$(\forall a)(\exists V, E)[|V| \geq a \wedge (V, E) \text{ is a degree } 2d \text{ graph} \\ \wedge(\forall U)(U \subseteq V \to \textit{edge-exp}_{(V,E)}(U) \geq 1/1296)].$$

In fact, there is a Σ^B_1-definable function G of VNC^1 so that that VNC^1 proves

$$(\forall a)[G(a) \text{ is a degree } 2d \text{ graph } G(a) = (V, E) \text{ with } |V| \geq a \\ \wedge(\forall U)(U \subseteq V \to \textit{edge-exp}_{(V,E)}(U) \geq 1/1296)].$$

VNC^1 can also prove the existence of edge expander graphs of aribitrary size.

▶ **Theorem 22.** *There is a constant $d = 2^\ell$ and a Σ^B_1-definable function G of VNC^1 so that VNC^1 proves*

$$(\forall a)[G(a) \text{ is a 4d-regular graph } G(a) = (V(a), E(a)) \text{ with } |V| = a \\ \wedge(\forall U)(U \subseteq V \to \textit{edge-exp}_{(V,E)}(U) \geq 1/(2 \cdot 1296))].$$

Proof. Pick appropriate constant values for d and c. VNC^1 starts by proving the existence of $\tilde{G}_i$ for the least i such that $2^i \geq a$. VNC^1 can prove the existence of the sequence $k_0, \ldots, k_s$ with $k_0 = i$, and each $k_{i+1} = \lfloor (k_i - 2c\ell - 5) \rfloor$ and s the first value with $k_s < 2c\ell + 7$. In addition, by Section 5.4, VNC^1 can prove the existence of second-order objects encoding edge expanders $\tilde{G}_j = (\tilde{V}_j, \tilde{E}_j)$ for every value $j = k_i$ or $j = k_i + 1$ with $i \leq s$. Recall that $|\tilde{V}_j| = 2^j$. Let $A(Y)$ express the condition that for some $i \leq s$, either (a) $|Y| = 2^{k_i} + 1$ and Y encodes a subset U of $\tilde{V}_{k_i}$ such that $\textit{edge-exp}_{\tilde{G}_{k_i}}(U) < 1/1296$, or (b) $|Y| = 2^{k_i+1} + 1$ and Y encodes a subset U of $\tilde{V}_{k_i+1}$ such that $\textit{edge-exp}_{\tilde{G}_{k_i+1}}(U) < 1/1296$. The (contrapositive) of the argument in Section 3.3, now shows that

$$(\exists Y{\leq}a)A(Y) \to (\exists Y{\leq}\sqrt{a})A(Y).$$

is VNC^1-provable. Applying Theorem 16 gives that VNC^1 proves

$$(\exists Y{\leq}a)A(Y) \to (\exists Y{\leq}1)A(Y).$$

Therefore, VNC^1 proves $\neg(\exists Y{\leq}a)A(Y)$, i.e., it proves the edge expansion properties for arbitrary Y, and hence the edge expansion properties of $\tilde{G}_i$. ◀

Finally, VNC^1 can also formalize the argument given in Section 4 to construct bipartite vertex expanders. The only new proof ingredient is the use of logarithms to define t_1 and t_2 in the proof of Claim 13. VNC^1 can define rational approximations to logarithms; here we need only integers t_1 and t_2 such that $(1 + \epsilon')^{t_1} \geq 1/\alpha$ and $(1 - \epsilon')^{t_2} \leq \alpha$. Since ϵ' is

small, these values can be estimated as $|\lceil 1/\alpha \rceil|/\epsilon'$. Actually, in the argument for Section 4, we have $\alpha = 1/600$ and $\epsilon' = \epsilon/D'$ are fixed constants; hence t_1 and t_2 are constants as well. Finally, at the very end of the proof of Theorem 12, we have $A = (D'(2d)+1)^{\max\{t_1,t_2\}}$, where $t' = \max\{t_1, t_2\}$. Thus A is also a constant. Here it is important that t' is constant, or at least is not too large, so that t' can be used as an exponent.

Thus we have proved the following theorem.

▶ **Theorem 23.** VNC^1 *proves Theorem 12 for any constant* α*. Namely, for any fixed rational* $0 < \alpha < 1$*, there exists an* $A > 0$ *and a* Σ^B_1*-defined function* $F(m)$ *of* VNC^1 *so that the following holds:* VNC^1 *proves that for all* m*,* $F(m)$ *equals the rotation map* Rot_G *of an* (α, A) *bipartite vertex expander graph* G *on* $m+m$ *vertices.*

As VNC^1 is a subtheory of VNC^1_*, Theorem 23 is stronger than the assumption needed by Jeřábek [23].

6 Application to monotone sequent calculus

In [36], Pudlák and Buss introduced a proof system for reasoning with monotone formulas, motivated by strong lower bounds results for monotone circuits, and posed the question whether similar difference in complexity holds in the propositional proof system setting. More specifically, they formulated monotone sequent calculus and asked whether any non-monotone proof of a monotone sequent can be replaced by a monotone proof at most polynomially larger. In [34], Pudlák further investigated this question, focusing in particular on the pigeonhole principle. There, he discussed the need to formalize properties of monotone counting formulas such as AKS sorting networks of [1], and asked whether there are small proofs of basic properties of counting formulas.

The pigeonhole principle was shown to have polynomial-size monotone sequent calculus proofs by Atserias, Galesi and Gavaldá in [6]; this paper was the first to use the name MLK for this system. The same paper also gave quasipolynomial-size proofs of basic counting principles. Building upon the latter result, Atserias, Galesi and Pudlák [7] show that, in contrast to monotone circuit classes, monotone proof systems are nearly as powerful as non-monotone ones: polynomial-size non-monotone proofs can be simulated by monotone ones of quasipolynomial size. The quasipolynomial blowup is introduced in the [6] proofs of certain properties of threshold formulas.

To prove that every LK proof can be converted into an MLK proof of quasipolynomial size, [7] use monotone threshold formulas to eliminate negated variables. A *threshold formula* $TH^n_k(x_1, \ldots, x_n)$ asserts that at least k variables x_i are 1. The standard inductive definition builds TH^n_k as a disjunction of $TH^{n/2}_i(x_1, \ldots, x_{n/2}) \wedge TH^{n/2}_j(x_{n/2+1}, \ldots, x_n)$ for all pairs $i, j \leq n/2$ such that $i + j \geq k$. This definition yields quasipolynomial size formulas TH^n_k, and thus gives only quasipolynomial size LK proofs of properties of TH^n_k. If LK is polynomially bounded, then so is MLK (as in this case properties of threshold functions would have polynomial-size LK proofs). More generally, they use the following lemma based on results from [6]:

▶ **Lemma 24** ([7, Lemma 6])**.** *Let* TH^n_k *be a polynomial-size monotone threshold formula. Then* MLK *polynomially simulates* LK *on monotone sequents, provided there are polynomial-size* LK *proofs of the following sequents:*

1. $TH^n_k(x_1, \ldots, x_n) \to$ *and* $\to TH^n_0(x_1, \ldots, x_n)$ *for every* n *and* $k > n$.
2. $TH^n_k(x_1, \ldots, x_i/0, \ldots, x_n) \to TH^n_{k+1}(x_1, \ldots, x_i/1, \ldots, x_n)$ *for all* n*,* k *and* i *such that* $0 \leq k, i \leq n$.

Such polynomial-size monotone threshold formulas can be built using the classic construction of monotone log-depth sorting networks by Ajtai, Komlós and Szemerédi [1], known as AKS sorting networks. A sorting network can be thought of as a circuit with n outputs gates, which contain the values of the input gates in sorted order. That is, the k^{th} output of a sorting network is 0 iff there are at least k 0s among inputs to the network. The construction of AKS sorting networks is fairly involved; see [31, 41] for expositions. At the end of the paper, Atserias et al. note that replacing their threshold formulas with monotone NC^1 sorting networks of Ajtai, Komlós and Szemerédi would remove the blowup and allow for *polynomial-size* simulation, provided the relevant properties can be proven with NC^1 reasoning (not necessarily monotone).

Jeřábek [24] has shown just that, under the assumption that bipartite expanders graphs with appropriate parameters can be constructed, and their properties proven in NC^1 reasoning. More precisely, Jeřábek [24] has shown that AKS sorting networks (Paterson's [31] variant) are indeed formalizable in a theory VNC^1_* of NC^1 reasoning, under the assumption of the existence of a family of bipartite expanders provable in VNC^1_* (with parameters as in Claim 13). The theory VNC^1_* is somewhat stronger than VNC^1 that we use, in that it can evaluate and reason about less uniform families of log-depth circuits; however, proofs in VNC^1_* still translate into polynomial-size LK proofs [23]. Thus, Jeřábek obtains the following result:

▶ **Theorem 25** ([24, Theorem 5.5]). *Suppose that there exists a constant D and a parameter-free NC^1_* function $G(m)$ such that VNC^1_* proves that for all numbers m, $G(m)$ is a $\langle 1/600, D\rangle$ bipartite $m+m$ expander. Then* MLK *polynomially simulates* LK *on monotone sequents.*

The construction in Theorem 12 gives expanders with the appropriate parameters, and Theorem 23 shows that it can be done in VNC^1 (and thus VNC^1_*). As this proves the assumption of Theorem 25, we immediately get the following corollary.

▶ **Theorem 26** (Main application). MLK *polynomially simulates* LK *on monotone sequents.*

7 Conclusions and open problems

From the point of view of bounded reverse mathematics, the area that tries to pinpoint the minimal reasoning power needed to prove mathematical theorems, it is very interesting to understand what is the complexity of reasoning required to prove properties of expander graphs, and thus what is the complexity of reasoning in expander-based proofs such as the known proofs of $\mathsf{SL} = \mathsf{L}$ [37, 39]. This paper makes a step in this direction by showing that an expander construction can be formalized within the system VNC^1.

A number of open questions remain. Can we formalize expanders in a weaker theory than VNC^1, e.g., the system of TC^0 reasoning? Can Reingold's result that undirected graph connectivity is in deterministic logspace [37] be formalized in the system of logspace reasoning? The analysis of graph powering given in this paper and the analysis of replacement product given in [4] are not strong enough to achieve that goal.

Finally, as was already asked by [24], can the AKS construction of expanders be modified to yield U_{E^*}-uniform sorting networks?

Acknowledgements. We want to thank Denis Thérien and Pascal Tesson for inviting V.K., A.K., and M.K. to the 2007 McGill Complexity Workshop in Barbados, where this paper was initiated. V.K. and A.K. also wish to thank Josh Buresh-Oppenheim, Shlomo Hoory, and Rahul Santhanam for our many discussions on expander graphs. V.K. and A.K. are particularly thankful to Russell Impagliazzo for inviting them to spend a semester at UCSD in

the spring of 2016, where this work was finally completed. S.B. thanks Amir Akbar Tabatabai and Raheleh Jalali for useful discussions on VNC[1], and Rosalie Iemhoff and Anupam Das for discussions on intuitionistic logic. We also thank Anupam Das for his comments on our paper, and Albert Atserias for clarifying to us the history of the MLK proof system. We are especially grateful to Emil Jeřábek for carefully reading our manuscript and pointing out some errors in the early versions.

References

1 Miklós Ajtai, Janós Komlós, and Endre Szemerédi. An $O(n \log n)$ sorting network. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, pages 1–9. Association for Computing Machinery, 1983.

2 Noga Alon and Fan R.K. Chung. Explicit construction of linear sized tolerant networks. *Discrete Mathematics*, 72:15–19, 1988.

3 Noga Alon and Yuval Roichman. Random Cayley graphs and expanders. *Random Structures and Algorithms*, 5:271–284, 1994.

4 Noga Alon, Oded Schwartz, and Asaf Shapira. An elementary construction of constant-degree expanders. *Comb. Probab. Comput.*, 17(3):319–327, May 2008. doi:10.1017/S0963548307008851.

5 Toshiyasu Arai. A bounded arithmetic AID for Frege systems. *Annals of Pure and Applied Logic*, 103:155–199, 2000.

6 Albert Atserias, Nicola Galesi, and Ricard Gavaldá. Monotone proofs of the pigeon hole principle. *Mathematical Logic Quarterly*, 47(4):461–474, 2001.

7 Albert Atserias, Nicola Galesi, and Pavel Pudlák. Monotone simulations of non-monotone proofs. *Journal of Computer and System Sciences*, 65(4):626–638, 2002. doi:10.1016/S0022-0000(02)00020-X.

8 Marta Bílková. Monotone sequent calculus and resolution. *Commentationes Mathematicae Universitatis Carolinae*, 42:575–582, 2001.

9 Sam Buss, Valentine Kabanets, Antonina Kolokolova, and Michal Koucký. Expander construction in VNC[1]. *Electronic Colloquium on Computational Complexity (ECCC)*, TR16-144, 2016. URL: http://eccc.hpi-web.de/report/2016/144/.

10 Samuel R. Buss. *Bounded Arithmetic*. Bibliopolis, 1986. Revision of 1985 Princeton University Ph.D. thesis.

11 Samuel R. Buss. Polynomial size proofs of the propositional pigeonhole principle. *Journal of Symbolic Logic*, 52:916–927, 1987.

12 Samuel R. Buss, Leszek Aleksander Kołodziejczyk, and Konrad Zdanowski. Collapsing modular counting in bounded arithmetic and constant depth propositional proofs. *Transactions of the AMS*, 367:7517–7563, 2015.

13 Peter Clote and Gaisi Takeuti. Bounded arithmetics for NC, ALOGTIME, L and NL. *Annals of Pure and Applied Logic*, 56:73–117, 1992.

14 Stephen Cook and Phuong Nguyen. *Logical Foundations of Proof Complexity*. Cambridge University Press, New York, NY, USA, 1st edition, 2010.

15 Stephen A. Cook. Feasibly constructive proofs and the propositional calculus. In *Proceedings of the Seventh Annual ACM Symposium on Theory of Computing*, pages 83–97, 1975.

16 Stephen A. Cook and Tsuyoshi Morioka. Quantified propositional calculus and a second-order theory for NC[1]. *Archive for Mathematical Logic*, 44:711–749, 2005.

17 Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3), June 2007. doi:10.1145/1236457.1236459.

18 Ofer Gabber and Zvi Galil. Explicit construction of linear sized superconcentrators. *Journal of Computer and System Sciences*, 22:407–420, 1981.

19 Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

20 Emil Jeřábek. Approximate counting in bounded arithmetic. *Journal of Symbolic Logic*, 72(3):959–993, 2007.

21 Emil Jeřábek. Approximate counting by hashing in bounded arithmetic. *Journal of Symbolic Logic*, 74(3):829–860, 2009.

22 Emil Jeřábek. Substitution frege and extended frege proof systems in non-classical logics. *Annals of Pure and Applied Logic*, 159(1):1 – 48, 2009. doi:10.1016/j.apal.2008.10.005.

23 Emil Jeřábek. On theories of bounded arithmetic for NC^1. *Annals of Pure and Applied Logic*, 162(4):322–340, 2011.

24 Emil Jeřábek. A sorting network in bounded arithmetic. *Annals of Pure and Applied Logic*, 162(4):341–355, 2011.

25 Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

26 Alexis Maciel, Toniann Pitassi, and Alan R. Woods. A new proof of the weak pigeonhole principle. *Journal of Computer and System Sciences*, 64(4):843–872, 2002.

27 Grigory Margulis. Explicit constructions of expanders. *Problems of Information Transmission*, pages 71–80, 1973.

28 Milena Mihail. Conductance and convergence of Markov chains: A combinatorial treatment of expanders. In *Proceedings of the Thirtieth Annual IEEE Symposium on Foundations of Computer Science*, pages 526–531, 1989.

29 Jeff B. Paris and Alex J. Wilkie. Δ_0 sets and induction. In W. Guzicki, W. Marek, A. Pelc, and C. Rauszer, editors, *Open Days in Model Theory and Set Theory*, pages 237–248, 1981.

30 Jeff B. Paris, Alex J. Wilkie, and A. R. Woods. Provability of the pigeonhole principle and the existence of infinitely many primes. *Journal of Symbolic Logic*, 53:1235–1244, 1988.

31 M. S. Paterson. Improved sorting networks with O(log N) depth. *Algorithmica*, 5(1-4):75–92, 1990. doi:10.1007/BF01840378.

32 Jan Pich. Logical strength of complexity theory and a formalization of the PCP theorem in bounded arithmetic. *Logical Methods in Computer Science*, 11(2:8):1–38, 2015.

33 Mark Pinsker. On the complexity of a concentrator. In *Proceedings of the Seventh Annual Teletraffic Conference*, pages 1–4, 1973.

34 P. Pudlak. On the complexity of the propositional calculus. In S. Barry Cooper and John K.Editors Truss, editors, *Sets and Proofs:*, London Mathematical Society Lecture Note Series, page 197–218. Cambridge University Press, Jun 1999.

35 Pavel Pudlák. Ramsey's theorem in bounded arithmetic. In *Computer Science Logic, Lecture Notes in Computer Science #553*, pages 308–312. Springer-Verlag, 1992.

36 Pavel Pudlák and Samuel R Buss. How to lie without being (easily) convicted and the lengths of proofs in propositional calculus. In *International Workshop on Computer Science Logic*, pages 151–162. Springer, 1994.

37 Omer Reingold. Undirected connectivity in log-space. *J. ACM*, 55(4):17:1–17:24, September 2008. doi:10.1145/1391289.1391291.

38 Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders. *Annals of Mathematics*, 155(1):157–187, 2002.

39 Eyal Rozenman and Salil P. Vadhan. Derandomized squaring of graphs. In *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th InternationalWorkshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings*, pages 436–447, 2005.

40 Walter L. Ruzzo. On uniform circuit complexity. *Journal of Computer and System Sciences*, 22:365–383, 1981.

41 Joel Seiferas. Sorting networks of logarithmic depth, further simplified. *Algorithmica (New York)*, 53(3):374–384, 2009. doi:10.1007/s00453-007-9025-6.

Finding Clearing Payments in Financial Networks with Credit Default Swaps is PPAD-complete*

Steffen Schuldenzucker[1], Sven Seuken[2], and Stefano Battiston[3]

1 **Department of Informatics, University of Zurich, Switzerland**
`schuldenzucker@ifi.uzh.ch`
2 **Department of Informatics, University of Zurich, Switzerland**
`seuken@ifi.uzh.ch`
3 **Department of Banking and Finance, University of Zurich, Switzerland**
`stefano.battiston@uzh.ch`

Abstract

We consider the problem of clearing a system of interconnected banks that have been exposed to a shock on their assets. Eisenberg and Noe [9] showed that when banks can only enter into simple debt contracts with each other, then a clearing vector of payments can be computed in polynomial time. In this paper, we show that the situation changes radically when banks can also enter into *credit default swaps* (*CDSs*), i.e., financial derivative contracts that depend on the default of another bank. We prove that computing an approximate solution to the clearing problem with sufficiently small constant error is PPAD-complete. To do this, we demonstrate how financial networks with debt and CDSs can encode arithmetic operations such as addition and multiplication. Our results have practical impact for network stress tests and reveal computational complexity as a new concern regarding the stability of the financial system.

1998 ACM Subject Classification J.4 [Computer Applications] Social and Behavioral Sciences – Economics

Keywords and phrases Financial Networks, Credit Default Swaps, Clearing Systems, Arithmetic Circuits, PPAD

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.32

1 Introduction

We consider systems of banks (or other financial institutions) that are connected by financial contracts. Due to a shock on their assets, some of the banks may not be able to meet their obligations towards other banks, thus forcing them into bankruptcy (or *default*). We study the *clearing problem* in this setting, i.e., the problem of computing a collection of payments between each pair of banks that are in accordance with standard bankruptcy law. Since banks' contractual relationships can be complex and are often cyclic, designing good clearing mechanisms is a nontrivial task.[1]

In their seminal paper, Eisenberg and Noe [9] devised an efficient clearing mechanism for financial systems. Their mechanism relies on the assumption that banks can only enter into simple *debt contracts*, i.e., loans from one bank to another. We argue, however, that the

* This project has received funding from the European Union's Horizon 2020 research and innovation programme under the DOLFINS project, grant agreement No 640772.

[1] We liberally borrow from our own earlier work [17] for parts of the introduction, related work, and the formal model.

© Steffen Schuldenzucker, Sven Seuken, and Stefano Battiston;
licensed under Creative Commons License CC-BY

growing importance of financial derivative contracts makes it necessary to reconsider the question if today's financial networks can still always be efficiently cleared. Specifically, *credit default swaps (CDSs)*, which are contracts that are only triggered when a reference entity goes into default, have received only little attention in a network context so far. Market participants use CDSs to insure themselves against a default of the reference entity or to place a speculative bet on this event. Because the reference entity can itself be a financial institution, CDSs create new dependencies that do not exist in pure debt networks.

In prior work [17], we have shown that if no money is lost in the bankruptcy process (i.e., banks do not incur *default costs*), then clearing payments always exist. However, our proof uses a non-constructive fixed point argument so that the question remained open if one could devise an efficient algorithm to actually *find* a clearing payment vector in this case.

In the present paper, we answer this question in the negative: we show that the problem FindClearing of finding an approximately clearing vector of payments in a financial network with debt and CDSs and without default costs is PPAD-complete even for a sufficiently small constant error bound. This implies that the problem does not have a polynomial-time approximation scheme (PTAS) unless P=PPAD and thus needs to be considered computationally intractable.

More in detail, we proceed as follows: we first describe a simplified variant of our model from [17] that only applies to the case without default costs (Section 3). We next argue that since solutions to the clearing problem can be irrational, FindClearing needs to be considered as an approximation problem. We define the notion of an *ε-approximately clearing vector* and we show that it makes the FindClearing problem well-posed and a member of PPAD (Section 4). Having done this, we describe our main contribution, namely a reduction from the problem of finding an approximate solution of a generalized circuit to FindClearing, establishing that FindClearing is PPAD-hard. We do this by composing *financial system gadgets*, i.e., fragments of financial networks that encode specific operations such as addition, subtraction, scaling, comparison, and Boolean operations like NOT and OR (Section 5).

Our results contribute to the literature on complexity in financial networks [3]. By studying financial networks from a computation perspective, we are able to accurately describe the effect of introducing a new class of financial products into the system in terms of computational complexity. Our hardness result has practical relevance for *stress tests*, in which regulators such as the European Central Bank evaluate the stability of the financial system under an array of adverse economic scenarios. We argue that, because of the complex interdependencies in real financial networks, future stress tests should take network effects into account, which would essentially require regulators to compute clearing payments. The approximation quality would be defined by the regulator and must thus be kept flexible. The fact that no PTAS for the clearing problem exists (unless P=PPAD) now implies that financial networks with CDSs cannot be reliably stress tested, which by itself poses a risk to the stability of the financial system.

2 Related Work

Prior work on financial networks has primarily focused on financial contagion, i.e., how small shocks may amplify to system-wide losses. Researchers have studied which network topologies are particularly susceptible to such effects [2, 10, 1] as well as developed measures for an individual bank's contribution to the risk of contagion [1, 4, 12].

The clearing problem was first studied by Eisenberg and Noe [9], who showed that in debt networks without default costs, clearing payments always exist and can be computed

in polynomial time. Rogers and Veraart [14] extended their result to debt networks *with* default costs.

Since all aforementioned pieces of work use a weighted graph as the underlying model of the financial network, they cannot accurately represent the *ternary* relationship introduced by a credit default swap between the holder, the writer, and the reference entity. We filled this gap in prior work [17] by devising a new model that *can* represent networks of debt and CDSs. We showed that the clearing problem in these networks is significantly more complex than in the debt-only case: if default costs are present, then clearing payments may not even exist and it is NP-hard to decide if they do. In the present paper, we study the case *without* default costs.

An extension of the clearing problem is to determine the maximum total loss an adversary could inflict on a financial system given a budget of shocks to banks. The problem is known [11] to be intractable in cross-ownership networks (which are similar to debt networks) despite the clearing problem being solvable in polynomial time.

The PPAD complexity class [13] is best known for the problem of computing a Nash equilibrium, the hardness of which was shown by reduction from generalized circuits [7, 5, 6, 15]. Our work builds on this technique, and in particular on Rubinstein's [15] PPAD-hardness result for constant accuracy. To the best of our knowledge, we are the first to implement generalized circuits using financial networks and, together with our prior work on the case with default costs, we are the first to present a computational complexity result for the clearing problem in financial networks.

3 Formal Model

Our model is based on the model by Eisenberg and Noe [9], which was restricted to debt contracts. We define an extension to credit default swaps. We adjust the notation where necessary. The model used in this paper is a simplified version of the one we previously introduced in [17], where default costs were also modeled. In this paper, we only consider financial systems without default costs.

We consider a two-period model:

- *Period 0:* Each bank receives an initial endowment called its *external assets*. Banks enter into bilateral contracts with each other. No bank is in default.
- *Period 1:* Banks' external assets change due to an exogenous shock. All banks must make payments according to their contractual commitments from period 0 and the new external assets.

We define the elements of the financial system in period 1.

Banks and External Assets

We denote by N a finite set of n *banks*. For any bank $i \in N$ let $e_i \geqslant 0$ denote the *external assets* of i as of period 1. Let $e = (e_i)_{i \in N}$ denote the vector of all external assets.

Contracts

There are two types of contracts: *debt contracts* and *credit default swap contracts (CDSs)*. Every contract gives rise to a conditional obligation to pay a certain amount, called a *liability*, from its *writer* to its *holder*. Banks that are unable to fulfill their obligations are said to be *in default*. The *recovery rate* r_i of a bank i is the share of its liabilities it is able to pay.

Thus, $r_i = 1$ if i is not in default and $r_i < 1$ if i is in default. Let $r = (r_i)_{i \in N}$ denote the vector of all recovery rates.

A *debt contract* obliges the writer i to unconditionally pay a certain amount to the holder j in period 1. This amount is called the *notional* of the contract and is denoted by $c^{\varnothing}_{i,j}$. A *credit default swap* obliges the writer i to make a conditional payment to the holder j in period 1. The amount of this payment depends on the default of a third bank k, called the *reference entity*. Specifically, the payment amount of the CDS contract from i to j with reference entity k and *notional* $c^k_{i,j}$ is $c^k_{i,j} \cdot (1 - r_k)$.

Note that when banks enter into contracts, there would typically be an initial payment. For example, debt contracts arise because the holder lends an amount of money to the writer, and holders of CDSs pay a premium to obtain them. In our model, any initial payments have been made in period 0 and are implicitly reflected by the external assets.

The contractual relationships between all banks are represented by a 3-dimensional matrix $c = (c^k_{i,j})_{i \in N,\, j \in N,\, k \in N \cup \{\varnothing\}}$. The entry $c^{\varnothing}_{i,j}$ is the total notional of the debt contracts from i to j and the entry $c^k_{i,j}$ for $k \in N$ is the total notional of CDS contracts from i to j with reference entity k. Zero entries indicate the absence of the respective contract. The set of contracts can alternatively be represented as an edge-weighted directed hypergraph.

We require that no bank enters a contract with itself (i.e., $c^k_{i,i} = 0$ for all $k \in N \cup \{\varnothing\}$ and $i \in N$). We further require that any bank that is a reference entity in a CDS must be a writer of some debt contract (i.e., for all $i \in N$, if $\sum_{k,l \in N} c^i_{k,l} > 0$, then $\sum_{j \in N} c^{\varnothing}_{i,j} > 0$). Both requirements are needed to rule out pathological cases. They are always assumed to hold in the following.

For any bank i, the *creditors of* i are the banks that are holders of contracts for which i is the writer, i.e., the banks to which i owes money. Conversely, the *debtors of* i are the writers of contracts of which i is the holder, i.e., the banks by which i is owed money. Note that the two sets can overlap: for example, a bank could hold a CDS on one reference entity while writing a CDS on another reference entity, both with the same counterparty.

Financial System Without Default Costs

A *financial system without default costs* (or, for the purpose of this paper just a *financial system*) is a tuple (N, e, c) where N is a set of banks, e is a vector of external assets, and c is a 3-dimensional matrix of contracts. The *length* of a financial system is the total number of bits needed to describe the tuple, including all numeric values.

Liabilities, Payments, and Assets

Given a recovery rate vector r, for any two banks i, j, the *liabilities of* i *to* j *at* r are the amount of money that i has to pay to j if recovery rates in the financial system are given by r, denoted by $l_{i,j}(r)$. They arise from the aggregate of all debt and CDS contracts from i to j:

$$l_{i,j}(r) := c^{\varnothing}_{i,j} + \sum_{k \in N} c^k_{i,j} \cdot (1 - r_k).$$

The *total liabilities of* i *at* r are the aggregate of all liabilities that i has towards other banks given the recovery rates r, denoted by $l_i(r)$:

$$l_i(r) := \sum_{j \in N} l_{i,j}(r).$$

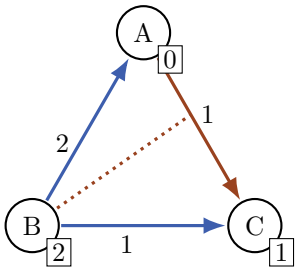


Figure 1 Example financial system.

The actual *payment* $p_{i,j}(r)$ from i to j at r can be lower than $l_{i,j}(r)$ if i is in default. By the *principle of proportionality*, a bank that is in default makes payments for its contracts in proportion to the respective liabilities:

$$p_{i,j}(r) := r_i \cdot l_{i,j}(r).$$

The *total assets* $a_i(r)$ of a bank i at r consist of its external assets e_i and the incoming payments to i:

$$a_i(r) := e_i + \sum_{j \in N} p_{j,i}(r).$$

Clearing Recovery Rate Vector

So far, r was just a candidate vector of recovery rates. We define what it means to be *clearing*:

▶ **Definition 3.1** (Clearing Recovery Rate Vector)**.** A recovery rate vector r is called *clearing* for a financial system without default costs $X = (N, e, c)$ if, for all banks $i \in N$, we have:

$$r_i = \min\left(1, \frac{a_i(r)}{l_i(r)}\right) \quad \text{if } l_i(r) > 0. \tag{1}$$

We also call a clearing recovery rate vector a *solution*.

Note that recovery rates of banks with zero liabilities are left unconstrained. While this might seem unintuitive at first, it corresponds exactly to our definition of the recovery rate: if there are no liabilities, then the "share of its liabilities bank i is able to pay" is not well-defined. Forcing the recovery rate to 1 in this case would introduce an artificial discontinuity because $\frac{a_i(r)}{l_i(r)}$ may converge to a value strictly below 1 while $l_i(r)$ converges to zero.[2,3]

[2] In the literature, (1) is often used directly as the definition of the recovery rate rather than "the share of its liabilities bank i is able to pay." Because we are always looking for a *clearing* recovery rate vector, the two definitions coincide.

[3] Eisenberg and Noe [9] define clearing *payments*, rather than recovery rates, by requiring that banks with sufficient assets pay their liabilities in full and banks without sufficient assets pay out all their assets proportionally to creditors. It is easy to show that this is equivalent to our definition.

Example and Visual Language

Figure 1 shows a visual representation of an example financial system. There are three banks $N = \{A, B, C\}$, drawn as circles, with external assets of $e_A = 0$, $e_B = 2$, and $e_C = 1$, drawn as rectangles on the banks. Debt contracts are drawn as blue arrows from the writer to the holder and they are annotated with the notionals $c^{\varnothing}_{B,A} = 2$ and $c^{\varnothing}_{B,C} = 1$. CDS contracts are drawn as orange arrows where a dashed line connects to the reference entity and are also annotated with notionals: $c^{B}_{A,C} = 1$. A clearing recovery rate vector for this example is given by $r_A = 1$, $r_B = \frac{2}{3}$, and $r_C = 1$. The liabilities arising from this recovery rate vector are $l_{B,A}(r) = 2$, $l_{B,C}(r) = 1$, and $l_{A,C}(r) = \frac{1}{3}$. The clearing payments are $p_{B,A}(r) = \frac{4}{3}$, $p_{B,C} = \frac{2}{3}$, and $p_{A,C}(r) = \frac{1}{3}$. This is the only solution for this system.

We stress that we are not concerned with the question whether or not it is "rational" for the banks to form a certain financial system: contracts might have been entered for reasons exogenous to the system or simply for cash transfers at time 0.

We are now ready to re-state an existence result which we have previously shown in [17].

▶ **Theorem 3.2** (Existence of Solutions [17]). *For every financial system without default costs, there exists a clearing recovery rate vector.*

Proof Outline. The proof rests on the fact that the right-hand side of equation (1) in Definition 3.1 is continuous as a function of r. Assume WLOG that $N = \{1, \ldots, n\}$. For $i \in N$ let

$$\rho_i : [0,1]^n \to 2^{[0,1]}$$
$$\rho_i(r) := \begin{cases} \{\min(1, \frac{a_i(r)}{l_i(r)})\} & \text{if } l_i(r) > 0 \\ [0,1] & \text{if } l_i(r) = 0 \end{cases}$$

and define $\rho : [0,1]^n \to 2^{[0,1]^n}$ by $\rho(r) := \times_{i=1}^{n} \rho_i(r)$.

Clearly, a recovery rate vector r is clearing iff it is a fixed point of the set-valued function ρ, i.e., iff $r \in \rho(r)$. To show that such a fixed point exists, one applies Kakutani's fixed point theorem for set-valued functions with a closed graph (a generalization of Brouwer's fixed point theorem for continuous functions). ◀

4 Defining the FindClearing Search Problem

We have just seen a non-constructive proof that a solution for a given financial system always exists. In this section, we define the corresponding total search problem. Since there are financial systems where all solutions contain irrational numbers (a simple example is provided in Appendix A), the best we can hope for is an algorithm that computes a recovery rate vector that is in some sense *approximately* clearing.

There are many ways to relax the definition of clearing recovery rate vectors to receive a concept of an approximate solution. The approach we will use in this paper is to relax the function ρ from the proof of Theorem 3.2. For $x \in \mathbb{R}$ let $[x] := \min(1, \max(0, x))$. For $\varepsilon \geqslant 0$ write $y = x \pm \varepsilon$ to mean that $|x - y| \leqslant \varepsilon$ if x and y are scalars and $\|x - y\| \leqslant \varepsilon$ if x and y are vectors, where $\|\cdot\|$ is the supremum norm. We also use the notation "$\pm\varepsilon$" in compound expressions such as $[x \pm \varepsilon]$ to indicate a range of possible values. This notation formally

corresponds to interval arithmetic. For $\varepsilon \geqslant 0$ and $i \in N = \{1, \ldots, n\}$ let

$$\rho_i^\varepsilon(r) : [0,1]^n \to 2^{[0,1]}$$
$$\rho_i^\varepsilon(r) := \begin{cases} [\frac{a_i(r)}{l_i(r)} \pm \varepsilon] & \text{if } l_i(r) > 0 \\ [0,1] & \text{if } l_i(r) = 0 \end{cases}$$

and let $\rho^\varepsilon : [0,1]^n \to [0,1]^n$ be defined accordingly.

▶ **Definition 4.1** (Approximately Clearing Recovery Rate Vector)**.** Fix a financial system without default costs and let $\varepsilon \geqslant 0$. A recovery rate vector r is called *ε-approximately clearing* or an *ε-solution* if it is a fixed point of the set-valued function ρ^ε, i.e., if $r \in \rho^\varepsilon(r)$. For clarity, we refer to solutions that are not approximate as *exact solutions*.

Our definition of an approximate solution has many desirable properties from an economic and technical point of view. We provide a discussion in Appendix B. Note in particular that if r is an ε-solution and $l_i(r) > 0$, then $r_i = [\frac{a_i(r)}{l_i(r)}] \pm \varepsilon$, though the converse does not necessarily hold.

It is easy to see that for any $\varepsilon > 0$, there always exists an ε-solution of finite length. To guarantee that there is also an ε-solution of *polynomial* length, we make an additional assumption that we call *non-degeneracy*.[4] We can then state our search problem.

▶ **Definition 4.2** (Non-degenerate Financial System)**.** A financial system without default costs $X = (N, e, c)$ is called *non-degenerate* if each bank that writes a CDS also writes a debt contract or has strictly positive external assets.

▶ **Definition 4.3** (ε-FindClearing Problem)**.** For any parameter $\varepsilon > 0$, ε-FindClearing is the following total search problem: given a non-degenerate financial system without default costs, find an ε-solution.

The following lemma establishes that under the assumption of non-degeneracy, sufficiently "short" approximate solutions always exist in the vicinity of exact solutions, thus making ε-FindClearing a well-posed search problem. The converse is not in general true: there can be additional approximate solutions that are not close to any exact solution. While this is unfortunate, it appears to be unavoidable for an approximate solution concept; for example, the well established concept of approximate Nash equilibrium also has this property.

▶ **Lemma 4.4** (ε-FindClearing is Well-posed and in PPAD)**.**

1. *If $X = (N, e, c)$ is a non-degenerate financial system without default costs and $\varepsilon > 0$, then there exists an ε-solution of length polynomial in the length of X and the length of ε.*
2. *For any $\varepsilon > 0$, the problem ε-FindClearing is in PPAD.*

Proof Outline (full proof in Appendix C). We define a function F such that any ε-approximate fixed point of F gives rise to an ε-solution of X. We prove that since X is non-degenerate, F has a polynomial Lipschitz constant. The lemma follows using standard techniques. ◀

5 FindClearing is PPAD-hard

Our main contribution in this paper is the proof that ε-FindClearing is PPAD-hard, and thus PPAD-complete, for a sufficiently small constant ε.

[4] It is an open question whether or not ε-solutions of polynomial length are also guaranteed to exist when this assumption is not made.

▶ **Theorem 5.1.** *There exists an $\varepsilon > 0$ such that the ε-FindClearing problem is PPAD-hard.*

The theorem immediately implies:

▶ **Corollary 5.2.** *There is no polynomial-time approximation scheme that computes an ε-solution for a given financial system without default costs and a given ε, unless $P = PPAD$.*

Towards a proof of the theorem, we proceed in two steps: we first introduce a variant of Rubinstein's [15] generalized circuit framework and we show that the problem of finding an approximate solution of a generalized circuit in this framework is still well-posed and PPAD-complete (Section 5.1). We then reduce this problem to ε-FindClearing (Section 5.2).

5.1 Generalized Circuits

A generalized circuit consists of a collection of interconnected arithmetic or Boolean gates. In contrast to regular arithmetic or Boolean circuits, generalized circuits may contain cycles, making the problem of finding a solution (or stable state) of the circuit a non-trivial fixed point problem. Rubinstein [15] introduced a framework for generalized circuits that is already well-suited for our purposes. To make our reduction to financial systems as simple as possible, we slightly adapt Rubinstein's definition by assuming a reduced set of gates.

▶ **Definition 5.3** (Generalized Circuit and Approximate Solution)**.** A *generalized circuit* is a collection of *nodes* and *gates*, where each node is labeled *input* of any number of gates (including zero) and *output* of at most one gate. Inputs to the same gate are distinguishable from each other. Each gate has one of the following types:

- For each $\zeta \in [0, 1]$ the *constant gate* C_ζ with no inputs and one output.
- Arithmetic gates: *addition* and *subtraction* gates, denoted C_+ and C_-, with two inputs and one output; for each $\zeta > 0$ the *scale by* ζ gate $C_{\times\zeta}$ with one input and one output.
- For each $\zeta \in (0, 1)$ the *compare to* ζ *gate* $C_{>\zeta}$ with one input and one output.
- Boolean gates: $C_\neg$ with one input and one output and $C_\vee$ with two inputs and one output.

The *length* of a generalized circuit is given by the number of nodes, the size of the mapping from nodes to inputs and outputs of gates, and the length of any ζ values involved.

If $\varepsilon \geqslant 0$ and C is a generalized circuit, then an *ε-approximate solution* (or *ε-solution*) to C is a mapping that assigns to each node v of C a value $x[v] \in [0, 1]$ such that at any gate of type g with inputs $a_1, \ldots, a_l$ and output v the respective condition from Figure 2 holds.

▶ **Definition 5.4** (ε-GCircuit Problem)**.** For any parameter $\varepsilon > 0$, ε-GCircuit is the following total search problem: given a generalized circuit, find an ε-solution.

Note how the comparison gadget $C_{>\zeta}$ is *brittle*: its value is arbitrary if $x[a_1]$ is close to ζ. This property is crucial for our second step of describing generalized circuits via financial systems because the function $\frac{a_i}{l_i}$ that ultimately defines an approximate solution is always continuous while a non-brittle comparison gadget, yielding low values for $x[a_1] < \zeta$ and high values for $x[a_1] \geqslant \zeta$, would correspond to a discontinuous function. We further use *approximate Boolean values* $0 \pm \varepsilon$ and $1 \pm \varepsilon$ instead of exact Boolean values 0 and 1 since the latter are not attainable if there can be ε errors at each bank. Note how chains of Boolean gadgets do not accumulate errors, but chains of arithmetic gadgets do.

It is well accepted in the literature that ε-GCircuit is well-posed and in PPAD. We provide a simple lemma for our variant of ε-GCircuit for completeness. PPAD-hardness, and thus PPAD-completeness, of the ε-GCircuit problem for constant ε follows by reduction from Rubinstein's variant. Both proofs can be found in Appendix D.

$$
\begin{aligned}
g = C_{\zeta} \quad &\Rightarrow \quad x[v] = \zeta \pm \varepsilon \\
g = C_{+} \quad &\Rightarrow \quad x[v] = [x[a_1] + x[a_2]] \pm \varepsilon \\
g = C_{-} \quad &\Rightarrow \quad x[v] = [x[a_1] - x[a_2]] \pm \varepsilon \\
g = C_{\times\zeta} \quad &\Rightarrow \quad x[v] = [\zeta \cdot x[a_1]] \pm (1+\zeta)\,\varepsilon \\
g = C_{>\zeta} \quad &\Rightarrow \quad x[a_1] \leqslant \zeta - \varepsilon \;\Rightarrow\; x[v] = 0 \pm \varepsilon \\
&\qquad\quad x[a_1] \geqslant \zeta + \varepsilon \;\Rightarrow\; x[v] = 1 \pm \varepsilon \\
g = C_{\neg} \quad &\Rightarrow \quad x[a_1] = 0 \pm \varepsilon \;\Rightarrow\; x[v] = 1 \pm \varepsilon \\
&\qquad\quad x[a_1] = 1 \pm \varepsilon \;\Rightarrow\; x[v] = 0 \pm \varepsilon \\
g = C_{\vee} \quad &\Rightarrow \quad x[a_1] = 0 \pm \varepsilon \text{ and } x[a_2] = 0 \pm \varepsilon \;\Rightarrow\; x[v] = 0 \pm \varepsilon \\
&\qquad\quad x[a_1] = 1 \pm \varepsilon \text{ or } \quad x[a_2] = 1 \pm \varepsilon \;\Rightarrow\; x[v] = 1 \pm \varepsilon
\end{aligned}
$$

Figure 2 Conditions to hold at the different gates in an ε-solution of a generalized circuit.

▶ **Lemma 5.5** (ε-GCircuit is Well-posed and in PPAD)**.**

1. *If C is a generalized circuit and $\varepsilon > 0$, then there exists an ε-solution for C of length polynomial in the length of C and the length of ε.*
2. *For any $\varepsilon > 0$, the ε-GCircuit problem is in PPAD.*

▶ **Lemma 5.6.** *There exists an $\varepsilon > 0$ such that the ε-GCircuit problem is PPAD-hard.*

5.2 Reduction from Generalized Circuits to Financial Systems

We now reduce the GCircuit problem to the FindClearing problem. To do so, we construct *financial system gadgets*, i.e., fragments of financial systems where the recovery rate of an *output bank* is given (approximately) by a function of certain *input banks*.

▶ **Definition 5.7** (Financial System Gadget)**.** A *financial system gadget* G is a polynomial-time computable function mapping a financial system without default costs $X = (N, e, c)$ to a new financial system $X' = (N', e', c')$ in the following way:

- Given are X, a set of *input banks* $a_1, \ldots, a_l \in N$ where l depends on the gadget, and an *output bank* $v \in N$ such that v has no assets or liabilities in X, i.e., $e_v = c^k_{v,j} = c^k_{j,v} = 0$ for all $j \in N$ and $k \in N \cup \{\varnothing\}$.
- X' consists of X together with some new banks and contracts.
- For any ε and any ε-solution r' of X', the restriction $r := r'|_N$ is an ε-solution for X.
- For any ε and any ε-solution r of X, there is an ε-solution r' of X' such that $r'_i = r_i$ for all $i \in N \backslash \{v\}$.

In addition to these properties, gadgets typically establish some relationship between the recovery rates of the input and output banks. We usually label input banks a and b instead of a_1 and a_2 for the sake of readability.

We will now describe our gadgets: addition gadgets, scaling and comparison gadgets, and Boolean gadgets. Some of the gadgets, shown in Figures 3–6, are *fundamental* while the others are defined as combinations of the fundamental ones. We use our graphical language for financial systems where we draw the (existing) input and output banks as dotted circles and the new banks as solid circles. Our gadgets add assets and liabilities to the output bank and CDS references to the input banks. This ensures that gadgets only restrict the recovery

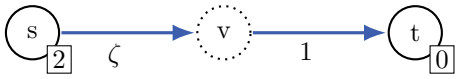


Figure 3 Constant Gadget: extension of an existing financial system with output bank v by new banks s, t and contracts such that $r_v = \zeta \pm \varepsilon$.

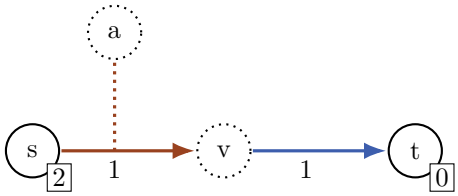


Figure 4 Inverter Gadget: extension of an existing financial system with input bank a and output bank v by new banks s, t and contracts such that $r_v = 1 - r_a \pm \varepsilon$.

rate of the output bank based on the recovery rates of the input banks, but not vice versa, and gadgets applied to different output banks do not conflict. In a final step, we iteratively apply our gadgets starting from a financial system with no contracts to receive a financial system that corresponds to a given generalized circuit. Our gadgets will be accurate up to an error of 3ε. We will later compensate for the factor 3 by choosing ε by factor 3 smaller. All gadgets lead to non-degenerate financial systems.

5.2.1 Addition Gadgets

The simplest gadget establishes a fixed recovery rate at the output bank:

▶ **Lemma 5.8** (Constant Gadget). *Let $\zeta \in [0, 1]$. There is a financial system gadget with no input banks and with output bank v such that if r is an ε-solution, then $r_v = \zeta \pm \varepsilon$.*

Proof. Consider the gadget in Figure 3. We have $\frac{a_s(r)}{l_s(r)} \geqslant 2 \geqslant 1 + \varepsilon$. It is easy to see that this implies that $r_s = 1$ in any ε-solution. Thus, s pays in full and $a_v(r) = \zeta$ and $l_v(r) = 1 \geqslant a_v(r)$, so in an ε-solution $r_v = \frac{a_v(r)}{l_v(r)} \pm \varepsilon = \zeta \pm \varepsilon$. ◀

An important building block for the following constructions is a gadget that "inverts" the recovery rate of a bank.

▶ **Lemma 5.9** (Inverter Gadget). *There is a financial system gadget with input bank a and output bank v such that if r is an ε-solution, then $r_v = 1 - r_a \pm \varepsilon$.*

Proof. Consider the gadget in Figure 4. Since $l_v(r) = 1$ we have in any ε-solution that $r_v = a_v(r) \pm \varepsilon$ and $a_v(r) = 1 - r_a$. ◀

We can now define the sum and difference gadgets:

▶ **Lemma 5.10** (Sum Gadget). *There is a financial system gadget with input banks a and b and output bank v such that if r is an ε-solution, then $r_v = [r_a + r_b] \pm 3\varepsilon$.*

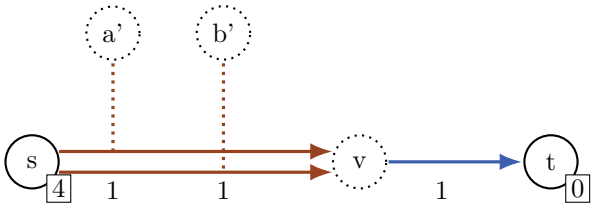


Figure 5 Sum Gadget: extension of an existing financial system with input banks a and b and output bank v by new banks s, t and contracts that translate $r_a + r_b$.

Proof. Apply inverter gadgets (Lemma 5.9) to both a and b and call the output banks a' and b', respectively. Now consider the gadget in Figure 5. We have

$$\begin{aligned} r_v &= [1 - r_{a'} + 1 - r_{b'}] \pm \varepsilon \\ &= [r_a + r_b \pm 2\varepsilon] \pm \varepsilon \\ &= [r_a + r_b] \pm 3\varepsilon. \end{aligned}$$

◀

▶ **Lemma 5.11** (Difference Gadget). *There is a financial system gadget with input banks a and b and output bank v such that if r is an ε-solution, then $r_v = [r_a - r_b] \pm 3\varepsilon$.*

Proof. Apply an inverter gadget (Lemma 5.9) to a and call the output bank a'. Apply the gadget in Figure 5 to a' and $b' := b$ and call the output bank u. From the proof of the previous lemma we know that

$$r_u = [1 - r_a + r_b] \pm 2\varepsilon$$

where the error is by one ε lower because we used one inverter gadget less. Now apply an inverter to u and call the output bank v. To show that r_v is as desired, we distinguish two cases:

- If $r_a \leqslant r_b$, then $1 - r_a + r_b \geqslant 1$, so $r_u = 1 \pm 2\varepsilon$ and thus $r_v = 1 - r_u \pm \varepsilon = 0 \pm 3\varepsilon = [r_a - r_b] \pm 3\varepsilon$ as required.
- If $r_a \geqslant r_b$, then $1 - r_a + r_b \leqslant 1$, so $r_u = 1 - r_a + r_b \pm 2\varepsilon$ and thus $r_v = 1 - r_u \pm \varepsilon = r_a - r_b \pm 3\varepsilon = [r_a - r_b] \pm 3\varepsilon$ as required.

◀

5.2.2 Scaling and Comparison

Towards the scaling and comparison gadgets, we introduce a versatile tool that can be used to re-scale and shift recovery rates.

▶ **Lemma 5.12** (Amplifier Gadget). *Let K and L be real numbers such that $K < L$, $K < 1$, and $L > 0$. Note that $K \leqslant 0$ and $L \geqslant 1$ are allowed. Let*

$$\begin{aligned} f &: [0,1] \to [0,1] \\ f(r_a) &:= \left[\frac{1}{L-K} r_a - \frac{K}{L-K} \right]. \end{aligned}$$

Note that f is monotonically increasing with $f(K) = 0$ and $f(L) = 1$.

There is a financial system gadget with input bank a and output bank v such that if r is an ε-solution, then $r_v = f(r_a) \pm (\delta + 1)\varepsilon$ where $\delta = \frac{1-K}{L-K}$. The construction can be performed in time polynomial in the lengths of L and K.

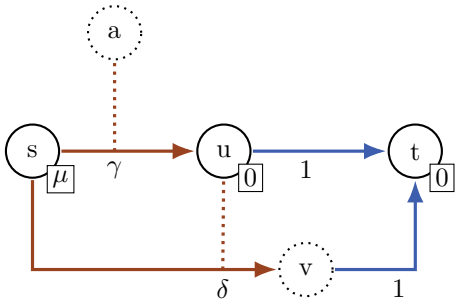


Figure 6 Amplifier Gadget: extension of an existing financial system with input bank a and output bank v by new banks s, t, u and contracts that translate the function f from Lemma 5.12. Let $\mu = 2(\gamma + \delta)$.

Proof. Consider the gadget in Figure 6 with

$$\gamma := \frac{1}{1-K}$$
$$\delta := \frac{1-K}{L-K}.$$

Let r be an ε-solution. We have

$$r_u = [\gamma(1 - r_a)] \pm \varepsilon$$
$$r_v = [\delta(1 - r_u)] \pm \varepsilon.$$

By replacing the first relation into the second one, we receive

$$\begin{aligned} r_v &\in [\delta\,(1 - ([\gamma(1 - r_a)] \pm \varepsilon))] \pm \varepsilon \\ &\subseteq [\delta\,(1 - [\gamma(1 - r_a)])] \pm (\delta + 1)\,\varepsilon \\ &= [\delta\,(1 - (\gamma(1 - r_a)))] \pm (\delta + 1)\,\varepsilon \\ &= [\delta - \delta\gamma + \delta\gamma r_a] \pm (\delta + 1)\,\varepsilon \;=\; \left[-\frac{K}{L-K} + \frac{1}{L-K} r_a\right] \pm (\delta + 1)\,\varepsilon \end{aligned}$$

where the third line is because $[\delta(1 - z)] = [\delta(1 - [z])]$ for any $z \geqslant 0$ and the last line is by simple algebra. Thus, r_v is as desired. ◀

We receive a scaling gadget by choosing $K = 0$:

▶ **Corollary 5.13** (Scale by Constant Gadget). *Let $\zeta > 0$. There is a financial system gadget with input bank a and output bank v such that if r is an ε-solution, then $r_v = [\zeta r_a] \pm (1 + \zeta)\varepsilon$. The construction can be performed in time polynomial in the length of ζ.*

Proof. Use an amplifier gadget (Lemma 5.12) with $K = 0$ and $L = \frac{1}{\zeta}$. Then $f(r_a) = [\zeta r_a]$ and $\delta = \zeta$. ◀

We receive a gadget that acts like the brittle comparison gate $C_{>\zeta}$ by choosing K and L closely together around a central point ζ. The gadget is less "brittle" the closer K and L are together, but this also increases the value δ and thus the output error of the gadget. To compensate for this, we first introduce a gadget that converts a wide range of values to approximate Boolean values with threshold 3ε.

▶ **Corollary 5.14** (Reset Gadget). *There is a financial system gadget with input bank a and output bank v such that if r is an ε-solution, then if $r_a \leqslant \frac{1}{4}$, then $r_v = 0 \pm 3\varepsilon$ and if $r_a \geqslant \frac{3}{4}$, then $r_v = 1 \pm 3\varepsilon$.*

Proof. Apply the amplifier gadget (Lemma 5.12) with $K = \frac{1}{4}$ and $L = \frac{3}{4}$. We have $\delta + 1 = \frac{5}{2} < 3$. If $r_a \leqslant \frac{1}{4}$, then $f(r_a) = 0$, so $r_v = f(r_a) \pm (1+\delta)\varepsilon = 1 \pm 3\varepsilon$. Likewise for $r_a \geqslant \frac{3}{4}$. ◀

▶ **Corollary 5.15** (Brittle Comparison to Constant Gadget). *Let $\zeta \in [0,1]$. There is a financial system gadget with input bank a and output bank v such that if $\varepsilon \leqslant 1/18$ and r is an ε-solution, then if $r_a \leqslant \zeta - 3\varepsilon$, then $r_v = 0 \pm 3\varepsilon$ and if $r_a \geqslant \zeta + 3\varepsilon$, then $r_v = 1 \pm 3\varepsilon$. The construction can be performed in time polynomial in the length of ζ.*

Proof. We apply two constructions involving the amplifier gadget (Lemma 5.12): first we apply an amplifier to a as an input bank with $K := \zeta - 3\varepsilon$ and $L := \zeta + 3\varepsilon$. Call the output bank u. We have $\delta = \frac{1-K}{L-K} = \frac{1-\zeta+3\varepsilon}{6\varepsilon} \leqslant \frac{1+3\varepsilon}{6\varepsilon} = \frac{1}{6\varepsilon} + \frac{1}{2}$. So this gadget has output error $(\delta+1)\varepsilon \leqslant \frac{1}{6} + \frac{1}{2}\varepsilon + \varepsilon \leqslant \frac{1}{4}$. Thus, if $r_a \leqslant K$, then $r_u \leqslant \frac{1}{4}$ and if $r_a \geqslant L$, then $r_u \geqslant \frac{3}{4}$. Now apply a reset gadget (Corollary 5.14) to u as the input bank to receive the desired lower output error of 3ε. ◀

5.2.3 Boolean Gadgets

We can re-use the addition gadgets from above to build Boolean gadgets, translating OR into "+" and NOT into "$1 - x$" (inversion). We use the reset gadget to prevent errors from propagating.

▶ **Lemma 5.16** (Boolean Gadgets). *There are financial system gadgets with input banks a and b and output bank v such that if $\varepsilon \leqslant 1/36$ and r is an ε-solution, then*

1. *(OR) If $r_a = 0 \pm 3\varepsilon$ and $r_b = 0 \pm 3\varepsilon$, then $r_v = 0 \pm 3\varepsilon$.*
   If $r_a = 1 \pm 3\varepsilon$ or $r_b = 1 \pm 3\varepsilon$, then $r_v = 1 \pm 3\varepsilon$.
2. *(NOT) If $r_a = 0 \pm 3\varepsilon$, then $r_v = 1 \pm 3\varepsilon$.*
   If $r_a = 1 \pm 3\varepsilon$, then $r_v = 0 \pm 3\varepsilon$.

Proof. 1. Apply a sum gadget (Lemma 5.10) to a and b and call the output bank u. Now apply a reset gadget (Corollary 5.14) to u and call the output bank v. We know that $r_u = [r_a + r_b] \pm 3\varepsilon$. If $r_a \geqslant 1 - 3\varepsilon$ or $r_b \geqslant 1 - 3\varepsilon$, then $r_u \geqslant 1 - 6\varepsilon \geqslant \frac{3}{4}$, so $r_v = 1 \pm 3\varepsilon$. If $r_a, r_b \leqslant 3\varepsilon$, then $r_u \leqslant 9\varepsilon \leqslant \frac{1}{4}$, so $r_v = 0 \pm 3\varepsilon$.

2. Apply similarly an inverter gadget (Lemma 5.9) and then a reset gadget. It is easy to show that the construction behaves as desired. ◀

5.2.4 Completing the PPAD-hardness Proof

We combine our gadgets to model generalized circuits, thus reducing ε-GCircuit to ε'-FindClearing (with $0 < \varepsilon' < \varepsilon$) and proving PPAD-hardness of ε-FindClearing:

Proof of Theorem 5.1. Let $\varepsilon > 0$ be arbitrary. We reduce ε-GCircuit to ε'-FindClearing where $\varepsilon' := \frac{\varepsilon}{3}$. Assume that we are given a generalized circuit C with n nodes and m gates. Construct a financial system via the following algorithm.

- Start with a system X^0 consisting of n banks, 0 external assets for each bank, and no contracts. Identify the n banks with the nodes of C.
- Consider the gates of C in any order. For each $t = 1, \ldots, m$ do the following:
  - Consider the t-th gate of C. Let g be the type, $a_1, \ldots, a_l$ the inputs, and v the output of this gate.

- Apply the gadget from above corresponding to g to X^{t-1} with input banks $a_1, \ldots, a_l$ and output bank v. Call the resulting financial system X^t.

- Let $X := X^m$.

For $t = 0, \ldots, m$ let C^t be C restricted to the first t gates. We show by induction on t that the ε'-solutions of X^t correspond to ε-solutions of C^t. For $t = 0$, the statement is clear. For $t > 0$, and assuming the statement for $t - 1$, it follows from the fact that the bank corresponding to the output of the t-th gate has no assets or liabilities in X^{t-1} and then from the definition of a financial system gadget and our above lemmas. By definition of the gadgets, each X^t, and thus X, is non-degenerate. ◀

▶ Remark. The intermediate systems X^t in the above construction may violate our assumption that any bank that is a reference entity in a CDS must be a writer of some debt contract (cf. Section 3). This happens when gadgets refer to a reference entity that is an output bank of another gadget that has not yet been executed. We can circumvent this problem by temporarily replacing such banks by a financial sub-system that fulfills all our assumptions and in which one of the banks can attain any recovery rate in some solution.[5] Alternatively, it is easy to show that not having the assumption does not lead to any problems in the proof.

6 Conclusion

In this paper, we have studied the problem of computing clearing payments in financial networks with debt and credit default swap (CDS) contracts and without default costs. We have shown that compared to debt-only networks, the addition of CDSs turns the clearing problem from being solvable exactly in polynomial time into an approximation problem that is PPAD-complete even when the desired approximation quality is kept constant. Consequently, no polynomial-time approximation scheme exists unless P=PPAD.

Further analysis shows that even very simple classes of financial systems can exhibit PPAD-hardness as long as banks are allowed to hold CDSs in a *naked* fashion, i.e., without also holding a corresponding debt contract from the reference entity.[6] Note that all our gadgets use naked CDSs, and they also seem to require them. Given this, future work should investigate whether financial networks in which naked CDSs are banned admit a polynomial-time algorithm for the clearing problem, similar to debt-only networks. We conjecture that this is the case. Another important task for future research is to find algorithms for general financial networks with CDSs that may not have polynomial worst-case running time, but are fast in practice. These algorithms could work by successively updating the set of defaulting banks in a systematic fashion. All algorithms for realistic financial systems must in addition be able to deal with non-linearities in the function $\frac{a_i}{l_i}$.

Acknowledgments. We would like to thank (in alphabetical order) Vitor Bosshard, Gianluca Brero, Yu Cheng, Marc Chesney, Constantinos Daskalakis, Marco d'Errico, Timo Mennle, Thomas Noe, and Joseph Stiglitz for helpful comments on this work. Furthermore, we are thankful for the feedback received from the anonymous reviewers at ITCS 2017. Any errors remain our own.

[5] Such a financial system is described in [17, Figure 3, $\delta = \gamma = 1$].

[6] We omit the analysis here due to space constraints. The interested reader is referred to our working paper [18]. For a discussion of naked CDSs, cf. Appendix B and [17].

References

1 Daron Acemoglu, Asuman Ozdaglar, and Alireza Tahbaz-Salehi. Systemic risk and stability in financial networks. *American Economic Review*, 105(2):564–608, Feb 2015.

2 Franklin Allen and Douglas Gale. Financial contagion. *Journal of political economy*, 108(1):1–33, 2000.

3 Stefano Battiston, J. Doyne Farmer, Andreas Flache, Diego Garlaschelli, Andrew G. Haldane, Hans Heesterbeek, Cars Hommes, Carlo Jaeger, Robert May, and Marten Scheffer. Complexity theory and financial regulation. *Science*, 351(6275):818–819, 2016. doi:10.1126/science.aad0299.

4 Stefano Battiston, Michelangelo Puliga, Rahul Kaushik, Paolo Tasca, and Guido Caldarelli. Debtrank: Too central to fail? financial networks, the fed and systemic risk. *Scientific reports*, 2, 2012.

5 X. Chen, X. Deng, and S. h. Teng. Computing nash equilibria: Approximation and smoothed complexity. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 603–612, Oct 2006. doi:10.1109/FOCS.2006.20.

6 Constantinos Daskalakis. On the complexity of approximating a nash equilibrium. *ACM Transactions on Algorithms (TALG)*, 2013. Special Issue for SODA 2011, Invited.

7 Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a nash equilibrium. *Electronic Colloquium on Computational Complexity (ECCC)*, 2005.

8 Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a nash equilibrium. *Commun. ACM*, 52(2):89–97, feb 2009. doi:10.1145/1461928.1461951.

9 Larry Eisenberg and Thomas H Noe. Systemic risk in financial systems. *Management Science*, 47(2):236–249, 2001.

10 Matthew Elliott, Benjamin Golub, and Matthew O. Jackson. Financial networks and contagion. *American Economic Review*, 104(10):3115–53, 2014. doi:10.1257/aer.104.10.3115.

11 Brett Hemenway and Sanjeev Khanna. Sensitivity and computational complexity in financial networks. Working Paper, Mar 2015. URL: http://arxiv.org/abs/1503.07676.

12 Daning Hu, J Leon Zhao, Zhimin Hua, and Michael CS Wong. Network-based modeling and analysis of systemic risk in banking systems. *MIS Quarterly*, 36(4):1269–1291, 2012.

13 Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498 – 532, 1994. doi:10.1016/S0022-0000(05)80063-7.

14 LCG Rogers and Luitgard AM Veraart. Failure and rescue in an interbank network. *Management Science*, 59(4):882–898, 2013.

15 Aviad Rubinstein. Inapproximability of nash equilibrium. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, STOC '15, pages 409–418, Portland, Oregon, USA, 2015. ACM. doi:10.1145/2746539.2746578.

16 Aviad Rubinstein. Inapproximability of nash equilibrium. Working Paper, 2016.

17 Steffen Schuldenzucker, Sven Seuken, and Stefano Battiston. Clearing payments in financial networks with credit default swaps. Extended abstract in *Proceedings of the 17th ACM Conference on Economics and Computation*, EC'16, Maastricht, The Netherlands, 2016. ACM. Current Working Paper: http://www.ifi.uzh.ch/ce/publications/Clearing_CDSs.pdf.

18 Steffen Schuldenzucker, Sven Seuken, and Stefano Battiston. Finding clearing payments in financial networks with credit default swaps is ppad-complete. Working Paper, 2016. URL: http://www.ifi.uzh.ch/ce/publications/Clearing_PPAD.pdf.

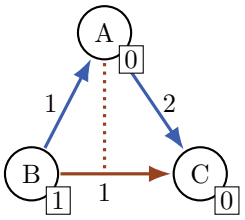


Figure 7 Financial System without default costs where the unique solution is irrational.

A Irrational Solutions

▶ **Example 1.1** (Irrational Solutions). Figure 7 shows a financial system the unique solution of which is irrational. To see this, note that by the contract structure r is clearing iff

$$r_A = \frac{1}{2} r_B, \qquad r_B = \frac{1}{2 - r_A},$$

and r_C is left unconstrained. One easily verifies that the unique solution in $[0,1]^2$ to this system of equations is given by

$$r_A = 2 - \sqrt{2}, \qquad r_B = 1 - \frac{1}{\sqrt{2}}.$$

B Properties of Approximate Solutions

Our definition of an approximate solution is well motivated from an economic point of view: assume that a bank A holds a debt contract of notional γ from bank B as well as a CDS on B from a highly capitalized bank C with the same notional. This contract pattern is called a *covered CDS* and it was the original use case CDSs were designed for: the CDS insures the debt contract. While nowadays, a large part of the CDSs are traded *naked* (i.e., they do not have this property), the covered case serves as a benchmark to which extent our solution concept is natural.

We describe the effect of ε errors in recovery rates on the three banks. If the insurer C is highly capitalized (its assets are greater than its liabilities by a factor $1 + \varepsilon$), then C never defaults ($r_C = 1$) and the assets of A are

$$\gamma r_B + \gamma(1 - r_B) r_C = \gamma.$$

That is, the covered CDS acts as a "full" insurance that eliminates A's dependence on B. This property is not affected by ε errors in the recovery rates of any bank. On the other hand, the writer C of the CDS might incur higher or lower liabilities due to errors in r_B, but this difference is bounded by $\varepsilon\gamma$. Finally, the recovery rate of B might be up to ε lower or higher than $\frac{a_B(r)}{l_B(r)}$. If it is lower, then B may keep up to $\varepsilon\gamma$ of its assets even though it is in default. If it is higher however, then B must make up to $\varepsilon\gamma$ in payments from money it does not have. This money would have to come from an external entity such as a government institution or the clearing mechanism itself. This is why clearing mechanisms should seek ε-solutions where ε is small compared to the inverse notionals in the system.

The following elementary properties serve as an indication that our definition of an approximate solution is also natural from a technical point of view. They are all easy to validate.

▶ **Proposition 2.1** (Natural Properties of Approximate Solutions). *Fix a financial system without default costs.*

1. *Any r is a 1-solution. r is a 0-solution iff it is an exact solution.*
2. *If $\varepsilon \leqslant \varepsilon'$, then any ε-solution is also an ε'-solution.*
3. *r is an ε-solution iff r is an ε'-solution for all $\varepsilon' > \varepsilon$.*
4. *Given r and ε, one can check in polynomial time if r is an ε-solution.*

C Proofs from Section 4

The following lemma lets us express ε-FindClearing as the problem of finding an approximate fixed point of a certain Lipschitz continuous function. Then the lemma follows using standard techniques.

▶ **Lemma C.1.** *Let $X = (N, e, c)$ be a non-degenerate financial system without default costs and let $\varepsilon > 0$. Assume WLOG that $N = \{1, \dots, n\}$. Define the function*

$$F : [0, 1+\varepsilon]^n \to [0, 1+\varepsilon]^n$$
$$F_i(s) := \begin{cases} \left[\frac{a_i([s])}{l_i([s])}\right]^{1+\varepsilon} & \text{if } l_i([s]) > 0 \\ 1+\varepsilon & \text{if } l_i([s]) = 0 \end{cases}$$

where $[x]^{1+\varepsilon} := \min(1+\varepsilon, \max(0, x))$ and $[s] := ([s_1], \dots, [s_n])$. Then the following hold:

1. *F is Lipschitz continuous with a Lipschitz constant polynomial-time computable from X and ε.*
2. *If s is an ε-approximate fixed point of F, then $[s]$ is an ε-solution of X.*

Proof. Part 1: It is sufficient to show that each F_i has an appropriate Lipschitz constant. So let $i \in N$. By non-degeneracy, bank i must fall into one of three cases: it either writes no contracts at all, or writes a debt contract, or has positive external assets. If i writes no contracts, then F_i is constant $1+\varepsilon$.

If i writes a debt contract, then $l_i([s]) > 0$ for all s, so

$$lF_i(s) = \left[\frac{a_i([s])}{l_i([s])}\right]^{1+\varepsilon} = \left([\,\cdot\,]^{1+\varepsilon} \circ \frac{a_i}{l_i} \circ [\,\cdot\,]\right)(s).$$

The functions $[\,\cdot\,]^{1+\varepsilon}$ and $[\,\cdot\,]$ are Lipschitz with constant 1. For $\frac{a_i}{l_i}$, we find a bound on the partial derivatives. We have

$$\begin{aligned} \frac{\partial \frac{a_i}{l_i}}{\partial r_k} &= \frac{\frac{\partial a_i}{\partial r_k} l_i - a_i \frac{\partial l_i}{\partial r_k}}{l_i^2} \\ &= \frac{(l_{k,i} - \sum_j r_j c^k_{j,i}) \cdot l_i + a_i \cdot \sum_j c^k_{i,j}}{l_i^2}. \end{aligned}$$

where the second line is easily seen by expanding a_i and l_i. The numerator is bounded from above in absolute value by

$$N^i_k := \left(c^{\varnothing}_{k,i} + \sum_j c^j_{k,i}\right) \cdot \left(\sum_j c^{\varnothing}_{i,j} + \sum_{j,l} c^l_{i,j}\right) + \left(e_i + \sum_j c^{\varnothing}_{j,i} + \sum_{j,l} c^l_{j,i}\right) \cdot \sum_j c^k_{j,i}$$

and the denominator is bounded from below by $D^i := (\sum_j c^{\varnothing}_{i,j})^2$. Thus, the partial derivative is bounded by $\frac{N^i_k}{D^i}$ and this bound is polynomial in X.

If i has positive external assets, then let $L_i := \{s \mid l_i([s]) > e_i\}$. For $s \notin L_i$, we have $F_i(s) = 1 + \varepsilon$ and further $F_i(s) \to 1 + \varepsilon$ as $l_i([s]) \to e_i$. On L_i, one receives a Lipschitz constant for the restriction of $\left[\frac{a_i([s])}{l_i([s])}\right]^{1+\varepsilon}$ to L_i by applying the same reasoning as above with $D_i := e_i^2$. Thus, F_i is the continuous union of two Lipschitz continuous functions and thus itself Lipschitz with the constant being the maximum of the two Lipschitz constants, namely $\max_k \frac{N_k^i}{D_i}$.

Part 2: Let s be an ε-approximate fixed point of F. Assume WLOG that $l_i([s]) > 0$. Let $i \in N$ and let $\tilde{s}_i := \frac{a_i([s])}{l_i([s])} \in [0, \infty)$. We have $F_i(s) = [\tilde{s}_i]^{1+\varepsilon}$ and $s_i = F_i(s) \pm \varepsilon$ and thus

$$\begin{aligned} & s_i = [\tilde{s}_i]^{1+\varepsilon} \pm \varepsilon \\ \Rightarrow\ & [s_i] \in \left[[\tilde{s}_i]^{1+\varepsilon} \pm \varepsilon\right] = [\tilde{s}_i \pm \varepsilon] \end{aligned}$$

where the last equality is easily seen by case distinction on $\tilde{s}_i \geqslant 1 + \varepsilon$ and $\tilde{s}_i < 1 + \varepsilon$. Thus, $[s]$ is an ε-solution at i. ◀

Proof of Lemma 4.4. Part 1: Let X and ε be given and consider the function F from Lemma C.1. Let K be the Lipschitz constant and recap that K is polynomial in X and ε. Since F is continuous on a compact domain, by Brouwer's fixed point theorem, it has an (exact) fixed point s. Let $\delta = \frac{\varepsilon}{K+1}$. Let s' be defined by $s'_i := \delta\lfloor\delta^{-1} s_i\rfloor$. That is, s'_i is s_i rounded to multiples of δ. s' has length $n \cdot L$ where L is the length of δ, and L is polynomial in the lengths of X and ε.[7] Further,

$$\begin{aligned} \left\|s' - F(s')\right\| &\leqslant \left\|s' - F(s)\right\| + \left\|F(s') - F(s)\right\| \\ &= \left\|s' - s\right\| + \left\|F(s') - F(s)\right\| \\ &\leqslant \delta + K\delta = (1 + K)\delta = \varepsilon. \end{aligned}$$

Hence, s' is an ε-approximate fixed point of F and thus an ε-solution.

Part 2: Proof by reduction to the PPAD-complete generic Brouwer problem [8]:

> Given an efficient algorithm for the evaluation of a function $F : [0,1]^n \to [0,1]^n$, a Lipschitz constant K for F, and an accuracy $\varepsilon > 0$, compute a point x such that $\|F(x) - x\| \leqslant \varepsilon$.

We apply the generic Brouwer problem to the function F from Lemma C.1. It is easy to see that one may replace the domain $[0,1]$ by $[0, 1+\varepsilon]$ without changing the problem in any significant way (e.g., by scaling inputs and outputs of F by a factor $1 + \varepsilon$ and replacing ε by $\frac{\varepsilon}{1+\varepsilon} \geqslant \frac{1}{2}\varepsilon$). Again by Lemma C.1, we know that the output of the Brouwer problem gives rise to an ε-solution for X. ◀

D Proofs from Section 5.1

Proof of Lemma 5.5. We show that the approximate solutions of a circuit correspond to the approximate fixed points of a certain Lipschitz continuous function. The statement of the lemma then follows like in the proof of Lemma 4.4.

[7] We assume here that numbers are encoded as fractions of binary integers. Alternatively, one could choose δ to be the largest power of two $\leqslant \frac{\varepsilon}{K+1}$.

For given C and ε define *gate functions* $f_g : [0,1]^l \to [0,1]$, where $l \in \{0,1,2\}$, as follows:

$$
\begin{aligned}
rLlf_{C_\zeta} &:= \zeta \\
f_{C_+}(a,b) &:= [a+b] \\
f_{C_-}(a,b) &:= [a-b] \\
f_{C_{\times\zeta}}(a) &:= [\zeta \cdot a] \\
f_{C_{>\zeta}}(a) &:= \left[\frac{1}{2\varepsilon}a + \frac{1}{2} - \frac{\zeta}{2\varepsilon}\right]
\end{aligned}
$$

Note that $f_{C_{>\zeta}}$ is monotonically increasing with $f_{C_{>\zeta}}(\zeta-\varepsilon) = 0$ and $f_{C_{>\zeta}}(\zeta+\varepsilon) = 1$. All gate functions are Lipschitz with constant $K := \max(2, \zeta_{\max}, \frac{1}{2\varepsilon})$ where $\zeta_{\max}$ is the maximum ζ such that C has a $C_{\times\zeta}$ gate.

Let $N = \{1, \dots, n\}$ be the set of nodes in the circuit. We define a function $F : [0,1]^n \to [0,1]^n$. For $x \in [0,1]^n$ and $i \in N$ let $F_i(x)$ be defined as follows:

- If i is an output of a gate g and the inputs of g are nodes $a_1, \dots, a_l$, then $F_i(x) := f_g(x_{a_1}, \dots, x_{a_l})$.
- If i is output of no gate, then $F_i(x) := x_i$.

Any ε-approximate fixed point of F is an ε-solution of C, though the converse does not hold. Since all gate functions are Lipschitz with constant K, so is F.

The first part of the lemma now follows just like in the proof of the first part of Lemma 4.4: if x is an exact fixed point of F and x' is x rounded to multiples of $\delta := \frac{\varepsilon}{K+1}$, then x' is an ε-approximate fixed point of F and thus an ε-solution of C and has polynomial length. It is not a problem that K depends on ε.

The second part of the lemma follows by reduction to the generic Brouwer problem just like in the proof of the second part of Lemma 4.4. This in fact proves that the weakly harder problem of computing an ε-solution where ε is not a parameter, but part of the input, is still in PPAD. It is again not a problem that K depends on ε because the generic Brouwer problem takes the Lipschitz constant as an input, just like ε. ◀

Proof of Lemma 5.6. Rubinstein [15] proved that the following variant of the ε-GCircuit problem is PPAD-hard for some ε:

1. Scaling is only allowed[8] by values $\zeta \leqslant 1$ and has error $\pm\varepsilon$ instead of $\pm(1+\zeta)\varepsilon$.
2. There are two additional, redundant gates: $C_=$ is a gate that (approximately) copies its input and $C_\wedge$ implements an approximate AND operator.
3. The comparison gate compares two inputs rather than compare one input to a constant.

For the first point, note that if $\zeta \leqslant 1$, then our $C_{\times\zeta}$ gate has error $(1+\zeta)\varepsilon \leqslant 2\varepsilon$ and thus we can achieve Rubinstein's error bound by considering an $\frac{\varepsilon}{2}$-solution instead. The second point does not make the problem any harder because we can express $C_=$ as $C_{\times 1}$ and $C_\wedge$ via the identity $x \wedge y = \neg(\neg x \vee \neg y)$.

Towards the third point, we show how to emulate the behavior of a binary comparison gate. Let a_1 and a_2 be the inputs and v the output of the would-be binary comparison gate. The expected behavior is that $x[v] = 0 \pm \varepsilon$ if $x[a_1] \leqslant x[a_2] - \varepsilon$ and $x[v] = 1 \pm \varepsilon$ if $x[a_1] \geqslant x[a_2] + \varepsilon$.

We rewrite the expression $x[a_1] < x[a_2]$ to use only comparison to a constant in a way that is robust against ε errors and cut-off at 0 and 1: construct, by combining the appropriate

[8] This assumption can be found in the full version of Rubinstein's paper [16].

gates, a sub-circuit corresponding to the expression $(\frac{1}{2} + (a_1 - a_2)) - (a_2 - a_1)$ and call the output node of that circuit u. If $\varepsilon' > 0$ and $x[\,\cdot\,]$ is an ε'-solution, then $x[u] = \tilde{u} \pm 4\varepsilon'$ where

$$\tilde{u} \;=\; \left[\left[\frac{1}{2} + [x[a_1] - x[a_2]]\right] - [x[a_2] - x[a_1]]\right] \;=\; \left[\frac{1}{2} + x[a_1] - x[a_2]\right].$$

Note that $x[a_1] < x[a_2] \Leftrightarrow \tilde{u} < \frac{1}{2}$. Add a $C_{>\frac{1}{2}}$ gate with input u and output v.

Now assume WLOG that $\varepsilon \leqslant \frac{1}{2}$, let $\varepsilon' = \frac{\varepsilon}{5}$, and let $x[\,\cdot\,]$ be an ε'-solution. Then

$$\begin{aligned} x[a_1] \leqslant x[a_2] - \varepsilon \Rightarrow \tilde{u} &\leqslant \frac{1}{2} - \varepsilon = \frac{1}{2} - 4\varepsilon' - \varepsilon' \\ &\Rightarrow x[u] \leqslant \frac{1}{2} - \varepsilon' \\ &\Rightarrow x[v] = 0 \pm \varepsilon' = 0 \pm \varepsilon. \end{aligned}$$

Analogously $x[a_1] \geqslant x[a_2] + \varepsilon \;\Rightarrow\; x[v] = 1 \pm \varepsilon$.

Altogether, we can construct from any circuit C in Rubinstein's [15] framework a circuit C' in our reduced framework such that the $\frac{\varepsilon}{5}$-solutions of C' are ε-solutions of C. This concludes the proof. ◀

Testing Submodularity and Other Properties of Valuation Functions

Eric Blais[*1] **and Abhinav Bommireddi**[2]

1 **David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada**
eric.blais@uwaterloo.ca

2 **David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada**
vabommir@uwaterloo.ca

Abstract

We show that for any constant $\epsilon > 0$ and $p \geq 1$, it is possible to distinguish functions $f : \{0,1\}^n \to [0,1]$ that are submodular from those that are ϵ-far from every submodular function in ℓ_p distance with a *constant* number of queries.

More generally, we extend the testing-by-implicit-learning framework of Diakonikolas et al. (2007) to show that every property of real-valued functions that is well-approximated in ℓ_2 distance by a class of k-juntas for some $k = O(1)$ can be tested in the ℓ_p-testing model with a constant number of queries. This result, combined with a recent junta theorem of Feldman and Vondrák (2016), yields the constant-query testability of submodularity. It also yields constant-query testing algorithms for a variety of other natural properties of valuation functions, including fractionally additive (XOS) functions, OXS functions, unit demand functions, coverage functions, and self-bounding functions.

1998 ACM Subject Classification F.2.0 Analysis of Algorithms and Problem Complexity

Keywords and phrases Property testing, Testing by implicit learning, Self-bounding functions

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.33

1 Introduction

Property testing is concerned with approximate decision problems of the following form: given oracle access to some function $f : \mathcal{X} \to \mathcal{Y}$ and some fixed property $\mathcal{P}$ of such functions, how many oracle calls (or queries) to f does a bounded-error randomized algorithm need to distinguish the cases where f has the property $\mathcal{P}$ from the case where f is ϵ-far—under some appropriately defined metric—from having the same property? Remarkably, many natural properties of functions can be tested with a number of queries that is *independent* of the size of the function's domain. For example, for any constant $\epsilon > 0$ and $t \geq 1$, a constant number of queries suffices to test whether a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is linear [7]; a polynomial of degree at most t [28]; a t-junta [5, 18]; a monomial [27]; computable by a Boolean circuit of size t [10]; or a linear threshold function [25].

In this work, we consider the problem of testing properties of bounded *real-valued* functions over the Boolean hypercube. In particular, are there natural examples of such properties that are testable with a constant number of queries? This question is best considered in the ℓ_p *testing* framework introduced by Berman, Raskhodnikova, and Yaroslavtsev [4]. In this

* E.B. is supported by an NSERC Discovery Grant.


© Eric Blais and Abhinav Bommireddi;
licensed under Creative Commons License CC-BY

setting, the distance between a function $f : \{0,1\}^n \to [0,1]$ and some property $\mathcal{P}$ of these functions is $\text{dist}_p(f, \mathcal{P}) = \inf_{g \in \mathcal{P}} \|f - g\|_p$.

1.1 Testing properties of valuation functions

Natural properties of bounded real-valued Boolean functions have been studied extensively in the context of valuation functions in algorithmic game theory. For a sequence of n goods labeled with the indices $1, \ldots, n$, we can encode the value of each subset of these goods to some agent with a function $f : \{0,1\}^n \to [0,1]$ by setting $f(x)$ to be the (possibly normalized) value of the subset $\{i \in [n] : x_i = 1\}$ to the agent. Such a valuation function f is

Additive if there are weights $w_1, \ldots, w_n$ such that $f(x) = \sum_{i:x_i=1} w_i$;

a **Coverage function** if there exists a universe U, non-negative weights $\{w_u\}_{u \in U}$, and subsets $A_1, ..., A_n \subseteq U$ such that $f(x) = \sum_{u \in \bigcup_{i:x_i=1} A_i} w_u$.

Unit demand if there are weights $w_1, \ldots, w_n$ such that $f(x) = \max\{w_i : x_i = 1\}$;

OXS if there are $k \geq 1$ unit demand functions $g_1, \ldots, g_k$ such that $f(x) = \max\{g_1(x^{(1)}), \ldots, g_k(x^{(k)})\}$ where the maximum is taken over all $x^{(1)}, \ldots, x^{(k)}$ such that for every $i \in [n]$, $x_i = \sum_{j=1}^{k} x_i^{(j)}$;

Gross Substitutes if for any $p' \leq p \in \mathbb{R}^n$ and any x, x' that maximize $f(x) - \sum_{i:x_i=1} p_i$ and $f(x') - \sum_{i:x'_i=1} p'_i$, respectively, every $j \in [n]$ for which $x_j = 1$ and $p_j = p'_j$ also satisfies $x'_j = 1$;

Submodular if $f(x) + f(y) \geq f(x \wedge y) + f(x \vee y)$ for every $x, y \in \{0,1\}^n$, where $\wedge$ and $\vee$ are the bitwise AND and OR operations;

Fractionally subadditive (XOS) iff there are non-negative real valued weights $\{w_{ij}\}_{i,j \leq n}$ such that $f(x) = \max_i \sum_j w_{ij} \cdot x_j$;

Self-bounding if $f(x) \geq \sum_i (f(x) - \min_{x_i} f(x))$, where $\min_{x_i} f(x) = \min\{f(x), f(x \oplus e_i)\}$ and $\oplus$ is the bitwise XOR operator; and

Subadditive if $f(x \cup y) \leq f(x) + f(y)$ for every $x, y \in \{0,1\}^n$.

Each of these properties enforces some structure on valuation functions, and much work has been devoted to better understanding these structures (and their algorithmic implications) by studying the properties through the lenses of learning theory [2,3,15], optimization [13,14], approximation [16, 17], and sketching [1]. The problem of testing whether an unknown valuation function satisfies one of these properties offers another angle from which we can learn more about the structure imposed on the functions that satisfy these properties.

Indeed, there has already been some recent developments on the study of testing these properties. Notably, Seshadhri and Vondrák [30] initiated the study of testing submodularity for functions over the hypercube and showed that in the setting where we measure the distance to submodularity in terms of Hamming distance (rather than ℓ_p distance), submodularity can be tested with $\epsilon^{-\sqrt{n}\log n}$ queries and that it *cannot* be tested with a number of queries that is independent of n. Subsequently, Feldman and Vondrák [17] showed that in the ℓ_1 testing framework, we can do much better: testing submodularity in this model requires a number of queries that is only logarithmic in n. Our first result shows that, in fact, for any value of $p \geq 1$, it is possible to test submodularity in the ℓ_p setting with a number of queries that is completely *independent* of n.

▶ **Theorem 1.1.** *For any $\epsilon > 0$ and any $p \geq 1$, there is an ϵ-tester for submodularity in the ℓ_p testing model with query complexity $2^{\tilde{O}(1/\epsilon^{\max\{2,p\}})}$.*

Another property that has been considered in the (standard Hamming distance) testing model is that of being a coverage function. Chakrabarty and Huang [8] showed that for

constant values of $\epsilon > 0$, $O(nm)$ queries suffice to ϵ-test whether a function f is a coverage function on some universe U of size $|U| \leq m$. Note that, unlike in the learning and approximation settings, bounds on the number of queries required to test some property $\mathcal{P}$ do not imply anything about number of queries required to test properties $\mathcal{P}' \subset \mathcal{P}$, so even though coverage functions are submodular, results on testing submodularity do not imply any bounds on the query complexity for testing coverage functions. Nonetheless, our next result shows that this property—along with most of the other properties of valuation functions listed above—can also be tested with a number of queries that is independent of n.

▶ **Theorem 1.2.** *For any $\epsilon > 0$ and any $p \geq 1$, there are ϵ-testers in the ℓ_p testing model for additive functions, coverage functions, unit demand functions, OXS functions, and gross substitute functions that each have query complexity $2^{\tilde{O}(1/\epsilon^{\max\{2,p\}})}$, and there are ϵ-testers in the ℓ_p testing model for fractional subadditivity and self-bounded functions that have query complexity $2^{2^{\tilde{O}(1/\epsilon^{\max\{2,p\}})}}$.*

Theorems 1.1 and 1.2 are both special cases of a general testing result that we obtain by extending the technique of *testing by implicit learning* of Diakonikolas et al. [10]. We describe this general result in more details below.

1.2 Testing real-valued functions by implicit learning

There is a strong connection between property testing and learning theory that goes back to the seminal work of Goldreich, Goldwasser, and Ron [21]. As they first observed, any proper learning algorithm for the class of functions that have some property $\mathcal{P}$ can also be used to test $\mathcal{P}$: run the learning algorithm, and verify whether the resulting hypothesis function h is close to the tested function f or not. This approach yields good bounds on the number of queries required to test many properties of functions, but, as simple information theory arguments show, it cannot yield query complexity bounds that are smaller than $\log n$ for almost all natural properties of functions over $\{0,1\}^n$.

Diakonikolas et al. [10] bypassed this limitation for the special case when every function that has some property $\mathcal{P}$ is close to a junta. A function $f : \{0,1\}^n \to [0,1]$ is a *k-junta* if there is a set $J \subseteq [n]$ of cardinality $|J| \leq k$ such that the value of f on any input x is completely determined by the values x_i for each $i \in J$. Every k-junta f has corresponding "core" functions $f_{\text{core}} : \{0,1\}^k \to \{0,1\}$ that define its value based on the value of the k relevant coordinates of its input. Diakonikolas et al.'s key insight is that for testing properties whose functions are (very) close to juntas, it suffices to learn the core of the input function—without having to identify the identity of the relevant coordinates.

The wide applicability of the testing-by-implicit-learning methodology is due to the fact that for many natural properties of Boolean functions, the functions that have these properties must necessarily be close to juntas under the Hamming distance. The starting point for the current research is a recent breakthrough of Feldman and Vondrák, who showed that a similar junta theorem holds for many properties of real-valued functions when closeness is measured according to ℓ_2 distance.

▶ **Feldman–Vondrák junta theorem.** *Fix any $\epsilon \in (0, \frac{1}{2})$. For every submodular function $f : \{0,1\}^n \to [0,1]$, there exists a submodular function $g : \{0,1\}^n \to [0,1]$ that is a $O(\frac{1}{\epsilon^2} \log \frac{1}{\epsilon})$-junta such that $\|f - g\|_2 \leq \epsilon$. Furthermore, for every self-bounding function $f : \{0,1\}^n \to [0,1]$, there exists a self-bounding function $g : \{0,1\}^n \to [0,1]$ that is a $2^{O(\frac{1}{\epsilon^2})}$-junta such that $\|f - g\|_2 \leq \epsilon$.*

The logarithmic dependence on n for the problem of testing submodularity in the ℓ_1 testing model [17] follows directly from Feldman and Vondrák's junta theorem and the (standard) testing-by-proper-learning connection. This junta theorem also suggests a natural approach for obtaining a constant query complexity for the same problem by combining it with a testing-by-implicit-learning algorithm. In order to implement this approach, however, new testing-by-implicit-learning techniques are required to overcome two obstacles.

The first obstacle is that most existing testing-by-implicit-learning algorithms [9–11, 22] are designed for properties that contain functions which are close to juntas in Hamming distance, not ℓ_p distance. This is a stronger condition, and enables the analysis of these algorithms to assume that with large probability, when f is very close to a k-junta f', the queries x made by the algorithm all satisfy $f(x) = f'(x)$. In the ℓ_p distance model, however, we can have a function f that is extremely close to a k-junta but still has $f(x) \neq f'(x)$ for many (or even every!) input x.

The second (related) obstacle that we encounter when considering submodular functions is that current testing-by-implicit-learning algorithms only work in the regime where the functions in $\mathcal{P}$ are ϵ-close to k-juntas for some $k < \epsilon^{-1/2}$. (See for example the discussion in §2.5 of [29].) This condition is satisfied by the properties of Boolean functions that have been studied previously, but the bounds in the Feldman–Vondrák junta theorem, however, do not satisfy this requirement.

We give a new algorithm for testing-by-implicit-learning that overcomes both of these obstacles. As a result, we obtain the following general theorem.

▶ **Theorem 1.3.** *For any $0 < \epsilon < \frac{1}{2}$ and any property $\mathcal{P}$ of functions mapping $\{0,1\}^n \to [0,1]$, if $k \geq 1$ is such that for every function $f \in \mathcal{P}$, there is a k-junta h that satisfies $\|f - h\|_2 \leq \frac{\epsilon}{10^6}$, then there is an ϵ-tester for $\mathcal{P}$ in the ℓ_2 testing model with query complexity $\frac{2^{O(k \log k)}}{\epsilon^{10}}$.*

Theorems 1.1 and 1.2 are both obtained directly from Theorem 1.3, the Feldman–Vondrák junta theorem and Fact 2.1.

▶ Remark. Another testing-by-implicit-learning algorithm, the Implicit Sieve algorithm of Wimmer and Yoshida [31] based on the Kushilevitz–Mansour learning algorithm [23], was introduced to test properties in the Hamming model but has a natural analogue in the ℓ_2 testing model as well. It would be quite interesting to determine what properties can be tested efficiently in the ℓ_p models by this algorithm (or extensions of it).

1.3 Overview of the proofs

1.3.1 The algorithm

The current testing-by-implicit-learning algorithms proceed in two main stages. In the first stage, the coordinates in $[n]$ are randomly partitioned into poly(k) parts, and an influence test is used to identify the (at most k) parts that contain relevant variables of an unknown input function f that is very close to being a k-junta. In the second stage, inputs $x \in [n]$ are drawn at random according to some distribution, the value $f(x)$ is observed, and the value of the relevant coordinate in each of the parts identified in the second stage is determined using more calls to the influence test.

The Implicit Learning Tester algorithm that we introduce in this paper reverses the order of the two main stages. In the first stage, it draws a sequence of q queries $X = (x^{(1)}, \ldots, x^{(q)})$ at random and queries the value of f on each of these queries. It also uses X to partition the coordinates in $[n]$ into 2^q random parts according to the values of the coordinate on the q queries. In the second stage, the algorithm then uses an influence

estimator to identify the k parts that contain the relevant coordinates of a k-junta that is close to f and, since all the coordinates in a common part have the same value on each of the q queries, learn the value of the k relevant coordinates on each of these initial queries. The algorithm then checks whether the core function thus learned is consistent with those of functions in the property being tested.

The main advantage of the Implicit Learning Tester algorithm that its analysis does not require the assumption that our samples are exactly consistent with those of an actual k-junta (instead of those of a function that is only promised to be *close* to a k-junta). This feature enables us to overcome the obstacles listed in the previous section, at the cost of adding a few complications to the analysis, as described below.

1.3.2 The analysis

There are two main technical ingredients in the analysis of the algorithm. The first, established in Lemma 3, is used to show that when f is close to a k-junta in ℓ_2 distance, the search procedure identifies parts that contain the k relevant coordinates of some k-junta that is close to f. (Note that the search is not guaranteed to find the parts that contain the relevant coordinates of the k-junta that is *closest* to f, but it suffices to find those of any close k-junta.)

The second technical ingredient addresses the fact that by drawing the q samples $x^{(1)}, \ldots, x^{(q)}$ first and then using these samples to provide the initial partition of the coordinates in $[n]$, we no longer will obtain uniformly random samples of the core f_{core} of the input function f. Nonetheless, in Lemma 4, we show that when f is close to a k-junta, the distribution of these samples on the core function still enables us to accurately estimate the distance of f_{core} to the core functions of any other k-junta.

1.4 Discussion and open problems

Theorems 1.1–1.3 raise a number of intriguing questions. The most obvious question left open is whether we can also test subadditivity of real-valued functions with a constant number of queries: subadditive functions need not be close to juntas, so such a result would appear to require a different technique.

It is also useful to compare our bounds for submodularity testing with those for testing monotonicity: in the Hamming distance testing model, Seshadhri and Vondrák [30] showed that the query complexity for testing submodularity is at least as large as that for testing monotonicity. However, the best current bounds for testing monotonicity in the ℓ_p testing model have a linear dependence on n [4]. Is it also possible to test monotonicity with a constant number of queries? Or is it the case that testing submodularity is strictly easier than testing monotonicity in the ℓ_p testing setting?

2 Preliminaries

Let $\mathcal{F}_n$ denote the set of functions mapping $\{0,1\}^n$ to $[0,1]$. For any $f \in \mathcal{F}_n$ and $S \subseteq [n]$ with complement $\overline{S} = [n] \setminus S$, when $x \in \{0,1\}^S$ and $y \in \{0,1\}^{\overline{S}}$, we write $f(x,y)$ to denote the value $f(z)$ for the input z that satisfies $z_i = x_i$ for each $i \in S$ and $z_i = y_i$ otherwise.

We use the standard definitions and notation for the Fourier analysis of functions $f : \{0,1\}^n \to [0,1]$. For a complete introduction to the topic, see [26]. Throughout the paper, unless otherwise specified all probabilities and expectations are over the uniform distribution on the random variable's domain.

2.1 Property testing

A *property* $\mathcal{P}$ of functions in $\mathcal{F}_n$ is a subset of these functions that is invariant under relabeling of the n coordinates. The *Hamming distance* between $f, g \in \mathcal{F}_n$ is $\mathrm{dist}_{\mathrm{Ham}}(f,g) = \Pr_x[f(x) \neq g(x)]$ and the Hamming distance between f and a property $\mathcal{P}$ is $\mathrm{dist}_{\mathrm{Ham}}(f,\mathcal{P}) = \inf_{g\in\mathcal{P}} \mathrm{dist}_{\mathrm{Ham}}(f,g)$. For $p \geq 1$, the *ℓ_p distance* between f and g is $\mathrm{dist}_p(f,g) = \|f-g\|_p = (\mathrm{E}_x[|f(x)-g(x)|^p])^{1/p}$ and the ℓ_p distance between f and $\mathcal{P}$ is $\mathrm{dist}_p(f,\mathcal{P}) = \inf_{g\in\mathcal{P}} \mathrm{dist}_p(f,g)$.

Given $\epsilon > 0$, An *ϵ-tester* in the Hamming testing model (resp., ℓ_p testing model) for some property $\mathcal{P} \subseteq \mathcal{F}_n$ is a randomized algorithm that (i) accepts every function $f \in \mathcal{P}$ with probability at least $\frac{2}{3}$ and (ii) rejects every function f that satisfies $\mathrm{dist}_{\mathrm{Ham}}(f,\mathcal{P}) \geq \epsilon$ (resp., $\mathrm{dist}_p(f,\mathcal{P}) \geq \epsilon$) with probability at least $\frac{2}{3}$. An ϵ-tester for $\mathcal{P}$ is an *(ϵ',ϵ)-tolerant tester*, for some $\epsilon' < \epsilon$ if it additionally accepts every function f that satisfies $\mathrm{dist}_{\mathrm{Ham}}(f,\mathcal{P}) \leq \epsilon'$ (resp., $\mathrm{dist}_p(f,\mathcal{P}) \leq \epsilon$) with probability at least $\frac{2}{3}$.

Our proofs of Theorems 1.1–1.3 are established in the ℓ_2 testing model. The result for general ℓ_p testing models is obtained from the following elementary relation between the query complexities of testing any property in different ℓ_p testing models.

▶ **Fact 2.1** (c.f. Fact 5.2 in [4]). *For any $\mathcal{P} \subseteq \mathcal{F}_n$, any $\epsilon > 0$, and any $p \geq 1$, the number $Q_p(\mathcal{P},\epsilon)$ of queries required to ϵ-test $\mathcal{P}$ in the ℓ_p testing model satisfies $Q_2(\mathcal{P},\epsilon) \leq Q_p(\mathcal{P},\epsilon) \leq Q_2(\mathcal{P},\epsilon^{\frac{p}{2}})$.*

Theorem 1.2 also relies on the following hierarchy of properties. (See, e.g., [24].)

▶ **Lemma 1.** *The properties of $\mathcal{F}_n$ defined in the introduction satisfy the inclusion hierarchy*

$$\begin{aligned}&\textit{Additive} \subseteq \textit{Coverage} \subseteq \textit{Unit demand} \subseteq \textit{OXS} \subseteq \textit{Gross substitute}\\ &\qquad \subseteq \textit{Submodularity} \subseteq \textit{XOS} \subseteq \textit{Self-bounding}.\end{aligned}$$

2.2 Juntas

The function $f : \{0,1\}^n \to [0,1]$ is a *junta on the set $J \subseteq [n]$* if for every $x, y \in \{0,1\}^n$ that satisfy $x_i = y_i$ for every $i \in J$, we have $f(x) = f(y)$. The function f is a *k-junta* if it is a junta on some set $J \subseteq [n]$ of cardinality $|J| \leq k$. The function $f_{\mathrm{core}} : \{0,1\}^k \to [0,1]$ is a *core function* of the k-junta $f : \{0,1\}^n \to [0,1]$ if there is a projection $\psi : \{0,1\}^n \to \{0,1\}^k$ defined by setting $\psi(x) = (x_{i_1},\ldots,x_{i_k})$ for some distinct $i_1,\ldots,i_k \in [n]$ such that for every $x \in \{0,1\}^n$, $f(x) = f_{\mathrm{core}}\big(\psi(x)\big)$.

▶ **Definition 2.2.** For any function $f : \{0,1\}^n \to [0,1]$ and set $J \subseteq [n]$, the *J-junta projection* of f is the function $f_J : \{0,1\}^J \to [0,1]$ defined by setting $f_J(x) = \mathrm{E}_{y\in\{0,1\}^{\overline{J}}}[f(x,y)]$ for every $x \in \{0,1\}^J$.

A basic fact that we will require is that f_J is the J-junta that is closest to f under the ℓ_2 metric.

▶ **Proposition 2.3.** *For every $f : \{0,1\}^n \to [0,1]$ and $J \subseteq [n]$, if $g : \{0,1\}^n \to [0,1]$ is a J-junta, then $\mathrm{dist}_2(f,f_J) \leq \mathrm{dist}_2(f,g)$.*

Proof. By applying the identity $\|f-g\|_2^2 = \|f-f_J+f_J-g\|_2^2$ and by expanding the right-hand side, we obtain

$$\begin{aligned}\|f-g\|_2^2 &= \mathop{\mathrm{E}}_{x\in\{0,1\}^J}\left[\mathop{\mathrm{E}}_{y\in\{0,1\}^{\overline{J}}}\Big[\big(f(x,y)-f_J(x,y)+f_J(x,y)-g(x,y)\big)^2\Big]\right]\\ &= \|f-f_J\|_2^2 + \|f_J-g\|_2^2 + 2\mathop{\mathrm{E}}_x\Big[\mathop{\mathrm{E}}_y\Big[\big(f(x,y)-f_J(x,y)\big)\big(f_J(x,y)-g(x,y)\big)\Big]\Big].\end{aligned}$$

Since $f_J - g$ is a J-junta, it does not depend on y and, by the definition of f_J, the last term equals 0. Therefore, $\|f - g\|_2^2 = \|f - f_J\|_2^2 + \|f_J - g\|_2^2$ and the claim follows. ◀

The property $\mathcal{P} \subseteq \mathcal{F}_n$ is a *property of k-juntas* if every function $f \in \mathcal{P}$ is a k-junta. The *core property* of a property $\mathcal{P}$ of k-juntas is the property $\mathcal{P}_{\text{core}} \subseteq \mathcal{F}_k$ defined by $\mathcal{P}_{\text{core}} = \{f_{\text{core}} : f \in \mathcal{P}\}$. For any $\gamma > 0$, the *γ-discretized approximation* of a function $f \in \mathcal{F}_n$ is the function $f^{(\gamma)}$ obtained by rounding the value $f(x)$ for each $x \in \{0,1\}^n$ to the nearest multiple of γ. The γ-discretized approximation of a property $\mathcal{P}$ is the property $\mathcal{P}^{(\gamma)} = \{f^{(\gamma)} : f \in \mathcal{P}\}$.

2.3 Influence

The notion of influence of coordinates in functions over the Boolean hypercube plays a central role in both our algorithm and its analysis. Informally, the influence of a set of coordinate measures how much re-randomizing these coordinates affects the value of the function. This notion is made precise as follows.

▶ **Definition 2.4.** The *influence* of a set $S \subseteq [n]$ of coordinates in the function $f : \{0,1\}^n \to [0,1]$ is

$$\mathrm{Inf}_f(S) := \mathop{\mathrm{E}}_{x \in \{0,1\}^{\overline{S}}}\Big[\mathop{\mathrm{Var}}_{y \in \{0,1\}^S} f(x,y)\Big] = \tfrac{1}{2}\mathop{\mathrm{E}}_{x \in \{0,1\}^{\overline{S}}}\Big[\mathop{\mathrm{E}}_{y,y' \in \{0,1\}^S}\Big[\big(f(x,y) - f(x,y')\big)^2\Big]\Big].$$

Our proofs make use of a few standard facts regarding the influence of sets of coordinates in f.

▶ **Fact 2.5.** *The influence of $S \subseteq [n]$ in $f \in \mathcal{F}_n$ is $\mathrm{Inf}_f(S) = \sum_{T : T \cap S \neq \emptyset} \hat{f}^2(T)$.*

▶ **Fact 2.6.** *For every $f \in \mathcal{F}_n$ and $S, T \subseteq [n]$, we have $\mathrm{Inf}_f(S) \leq \mathrm{Inf}_f(S \cup T) \leq \mathrm{Inf}_f(S) + \mathrm{Inf}_f(T)$.*

▶ **Fact 2.7.** *For every $f \in \mathcal{F}_n$ and $J \subseteq [n]$, we have $\mathrm{Inf}_f(\overline{J}) = \mathrm{dist}_2(f, f_J)^2$.*

▶ **Proposition 2.8.** *Fix $\epsilon > 0$, and let $f, g : \{0,1\}^n \to [0,1]$ satisfy $\mathrm{dist}_2(f,g) \leq \epsilon$. Then for any set $S \subseteq [n]$, $|\mathrm{Inf}_f(S)^{\frac{1}{2}} - \mathrm{Inf}_g(S)^{\frac{1}{2}}| \leq \epsilon$.*

Proof. By Fact 2.7, we have $\mathrm{Inf}_f(S)^{\frac{1}{2}} = \|f - f_{\overline{S}}\|_2$ and $\mathrm{Inf}_g(S)^{\frac{1}{2}} = \|g - g_{\overline{S}}\|_2$. By Proposition 2.3, we also have that $\|f - f_{\overline{S}}\|_2 \leq \|f - g_{\overline{S}}\|_2$. Combining these observations with the triangle inequality, we obtain $\mathrm{Inf}_f(S)^{\frac{1}{2}} - \mathrm{Inf}_g(S)^{\frac{1}{2}} = \|f - f_{\overline{S}}\|_2 - \|g - g_{\overline{S}}\|_2 \leq \|f - g_{\overline{S}}\|_2 - \|g - g_{\overline{S}}\|_2 \leq \|f - g\|_2 \leq \epsilon$. Hence $\mathrm{Inf}_f(S)^{\frac{1}{2}} - \mathrm{Inf}_g(S)^{\frac{1}{2}} \leq \epsilon$ and, similarly, $\mathrm{Inf}_g(S)^{\frac{1}{2}} - \mathrm{Inf}_f(S)^{\frac{1}{2}} \leq \epsilon$ as well. ◀

▶ **Proposition 2.9.** *There is an algorithm* EstimateInf *such that for every $f : \{0,1\}^n \to [0,1]$, $S \subseteq [n]$, $m \geq 1$, and $t \geq 0$, it makes m queries to f and returns an estimate of the influence of S in f that satisfies*

$$\Pr\big[|\mathrm{Inf}_f(S) - \textsc{EstimateInf}(f, S, m)| \geq t\big] \leq 2e^{-2mt^2}.$$

We also use the following key lemma from [6].

▶ **Lemma 2** (Lemma 2.3 in [6]). *Let $f : \{0,1\}^n \to [0,1]$ be a function that is ϵ-far from k-juntas and P be a random partition of $[n]$ into $r > 20k^2$ parts. Then with probability at least $\frac{5}{6}$, $\mathrm{Inf}_f(\overline{J}) \geq \frac{\epsilon^2}{4}$ for any union J of k parts from P.*

For the reader's convenience, we include the proof of Lemma 2 in Appendix A; though the original lemma in [6] was only for Boolean-valued functions, the proof remains essentially unchanged.

Algorithm 1: IMPLICIT LEARNING TESTER$(\mathcal{F}, k, \epsilon)$

Data: $q = \frac{2^{O(k)}}{\epsilon^5}$, $m = O(\frac{k^6}{\epsilon^5})$, $r = \log \frac{2^k}{100k^4}$

1 Draw $x^{(1)}, \ldots, x^{(q)} \in \{0,1\}^n$ independently and uniformly at random;

2 For each $c \in \{0,1\}^q$, define $S_c \leftarrow \left\{i \in [n] : \left(x_i^{(1)}, \ldots, x_i^{(q)}\right) = c\right\}$;

3 Let $P_1, \ldots, P_{100k^4}$ be a random equi-partition of $\{0,1\}^q$;

4 **for** *each* $J \subseteq [100k^4]$ *of size* $|J| = k$ **do**

5 $\quad S_J \leftarrow \bigcup_{j \in J} \bigcup_{c \in P_j} S_c$;

6 $\quad \eta_J \leftarrow \text{ESTIMATEINF}(f, [n] \setminus S_J, m)$;

7 $\{j_1^*, \ldots, j_k^*\} \leftarrow \text{argmin}_J \, \eta_J$;

8 $(P_{0,1}, \ldots, P_{0,k}) \leftarrow (P_{j_1^*}, \ldots, P_{j_k^*})$;

9 **for** $\ell = 1, \ldots, r$ **do**

10 $\quad$ Let $P_{\ell,i,0}, P_{\ell,i,1}$ be a random equi-partition of $P_{\ell-1,i}$ for each $i \le k$;

11 $\quad$ **for** *every* $z \in \{0,1\}^k$ **do**

12 $\quad\quad S_z \leftarrow \bigcup_{i \le k} \bigcup_{c \in P_{\ell,i,z_i}} S_c$;

13 $\quad\quad \eta_z \leftarrow \text{ESTIMATEINF}(f, [n] \setminus S_z, m)$;

14 $\quad z_\ell^* \leftarrow \text{argmin}_z \, \eta_z$;

15 $\quad$ For each $i \le k$, update $P_{\ell,i} \leftarrow P_{\ell,i,z_\ell^*}$;

16 Let $B = \{b_1, ..., b_k\} \leftarrow \bigcup_{i \le k} P_{r,i}$;

17 If $\text{ESTIMATEINF}(f, [n] \setminus S_B, m) > \epsilon^2/1000$, reject;

18 Let $\phi : \{0,1\}^n \to \{0,1\}^k$ be any projection that satisfies $\phi(x)_i \in S_{b_i}$ for each $i \le k$;

19 **for** $h \in \mathcal{F}_{\text{core}}^{(\frac{\epsilon}{1000})}$ **do**

20 $\quad$ If $\frac{1}{q} \sum_{i=1}^q \left(f(x^{(i)}) - h(\phi(x^{(i)}))\right)^2 \le 0.35\epsilon$, accept and return h;

21 Reject;

3 Testing by implicit learning

The proof of Theorem 1.3 is established by analyzing the IMPLICIT LEARNING TESTER algorithm.

3.1 Proof of Theorem 1.3

The analysis of the IMPLICIT LEARNING TESTER relies on two technical lemmas. The first shows that when the input function f is close to a k-junta, then with reasonably large probability, the function f is close to a junta on the set B of k parts that is identified by the algorithm.

▶ **Lemma 3.** *For any $\varepsilon > 0$, if the function $f : \{0,1\}^n \to [0,1]$ is ε-close to a k-junta and every call to* ESTIMATEINF *returns an influence estimate with additive error at most $\frac{\varepsilon^2}{100k^2}$, then the set B obtained by the* JUNTA-PROPERTY TESTER *satisfies* $\Pr\left[\text{Inf}_f([n] \setminus S_B) > 100\varepsilon^2\right] \le \frac{1}{20}$.

The second lemma shows that the estimate in Step 20 provides a good estimate of the distance between f and the functions in $\mathcal{P}$.

▶ **Lemma 4.** *Fix $\varepsilon > 0$. Let $f : \{0,1\}^n \to [0,1]$ be a function that satisfies $\text{dist}_2(f, g) \le \varepsilon$ for some function g that is a junta on $J \subseteq [n]$, $|J| \le k$. Then for every $h_{\text{core}} \in \mathcal{F}_{\text{core}}^{(\frac{\varepsilon}{1000})}$, the*

mapping $\psi : \{0,1\}^n \to \{0,1\}^k$ *defined in the* Implicit Learning Tester *and the function* $h = h_{\text{core}} \circ \psi$ *satisfy*

$$\left| \left(\tfrac{1}{q} \sum_{i}^{q} \left(f(x^{(i)}) - h(x^{(i)}) \right)^2 \right)^{\frac{1}{2}} - \text{dist}_2(g,h) \right| \leq 3\varepsilon$$

except with probability at most $2e^{-16q\varepsilon^4} + \frac{5k^2}{2^q}$.

The proofs of these lemmas are presented in Sections 3.2 and 3.3. We now show how they are used to complete the proof of Theorem 1.3.

As a first observation, we note that by Hoeffding's inequality and the union bound, all of the calls to EstimateInf have additive error at most $\frac{\epsilon^2}{10^6 k^2}$ except with probability at most $\frac{1}{6}$. In the following, we assume that this condition holds and show how, when it does, the algorithm correctly accepts or rejects with probability with probability at least $\frac{5}{6}$.

▶ **Claim 3.1** (Completeness). *When* f *is* $\frac{\epsilon}{10^6}$*-close to the property* $\mathcal{F}$ *of* k*-juntas, the* Implicit Learning Tester *accepts with probability at least* $\frac{5}{6}$.

Proof. First, by Lemma 3, the probability that f is rejected on step 17 is at most $\frac{1}{18}$. In the rest of the proof, we will show that except with probability at most $\frac{1}{9}$, there is a function $h_{\text{core}} \in \mathcal{F}_{\text{core}}^{(\frac{\epsilon}{1000})}$ for which the algorithm accepts on line 20.

Let $g \in \mathcal{F}$ be a function that satisfies $\text{dist}_2(f,g) \leq \frac{\epsilon}{10^6}$. Without loss of generality, we can assume that g is a junta on $[k]$. Let $J = [k] \cap S_B$ be the set of the junta variables of g that are contained in the final parts selected by the algorithm. Again without loss of generality (by relabeling the input variables once again if necessary), we can assume that $J = [j]$ for some $j \leq k$, and $i \in S_{b_i}$, for $i \leq j$.

Define $\psi : \{0,1\}^n \to \{0,1\}^k$ to be the mapping defined by $\psi(x) = (x_1, \ldots, x_j, x_{i_1}, \ldots, x_{i_{k-j}})$ where $i_1, \ldots, i_{k-j} \in [n] \setminus [k]$ are representative coordinates from the remaining parts $b \in B$ for which $P_b \cap [k] = \emptyset$.

Let $g_{\text{core}} \in \mathcal{F}_{\text{core}}$ be the core of g corresponding to the projection $\psi(x) = (x_1, \ldots, x_k)$, and let $h_{\text{core}} \in \mathcal{F}_{\text{core}}^{(\frac{\epsilon}{10^6})}$ be the discretized approximation to g_{core}. Define $h = h_{\text{core}} \circ \psi$. By our choice of g, we have $\text{dist}_2(f,g) \leq \frac{\epsilon}{10^6}$. In order to invoke Lemma 4, we now want to bound $\text{dist}_2(g,h)$.

Let $h^* \in \mathcal{F}^{(\frac{\epsilon}{10^6})}$, be the discretized approximation of g. Then $\text{dist}_2(g,h^*) \leq \frac{\epsilon}{10^6}$ and the triangle inequality implies that

$$\text{dist}_2(f,h^*) \leq \text{dist}_2(f,g) + \text{dist}_2(g,h^*) \leq \tfrac{2\epsilon}{10^6}$$

and that

$$\text{dist}_2(g,h) \leq \text{dist}_2(g,h^*) + \text{dist}_2(h^*,h) \leq \text{dist}_2(h^*,h) + \tfrac{\epsilon}{10^6}.$$

Furthermore, since $h_{\text{core}} = h^*_{\text{core}}$,

$$\begin{aligned} \text{dist}_2(h^*,h) &= \mathop{\mathrm{E}}_{x} \left[\left(h^*_{\text{core}}(x_1, \ldots, x_k) - h^*_{\text{core}}(x_1, \ldots, x_j, x_{i_1}, \ldots, x_{i_{k-j}}) \right)^2 \right]^{\frac{1}{2}} \\ &= 2\,\text{Inf}_{h^*_{\text{core}}}([k] \setminus [j])^{\frac{1}{2}} = 2\,\text{Inf}_{h^*}([n] \setminus [j])^{\frac{1}{2}}. \end{aligned}$$

By Proposition 2.8 and Lemma 3, except with probability at most $\frac{1}{18}$,

$$\text{Inf}_{h^*}([n] \setminus [j])^{\frac{1}{2}} \leq \text{Inf}_f([n] \setminus [j])^{\frac{1}{2}} + \text{dist}_2(f,h^*) \leq \text{Inf}_f([n] \setminus S_B)^{\frac{1}{2}} + \tfrac{2\epsilon}{10^6} \leq \tfrac{12\epsilon}{10^6}$$

and the distance between g and h is bounded by $\text{dist}_2(g,h) \leq \frac{13}{10^6}\epsilon$. When this bound holds, by Lemma 4 with $\varepsilon = \frac{\epsilon}{100}$, the algorithm accepts f for this h except with probability at most $\frac{1}{18}$. ◀

▶ **Claim 3.2** (Soundness I). *If f is $\frac{\epsilon}{100}$-far from being a k-junta, then the* Implicit Learning Tester *rejects with probability at least $\frac{5}{6}$.*

Proof. The initial partition $S_{P_1}, \ldots, S_{P_{100k^4}}$ is a random partition of $[n]$ with more than $20k^2$ parts so, by Lemma 2, with probability at least $\frac{5}{6}$, for any union $J \subseteq [n]$ of at most k of these parts we have $\mathrm{Inf}_f([n] \setminus J) \geq \frac{\epsilon^2}{400}$. When this is the case, the inclusion $S_B \subseteq \overline{L_0}$ and the fact that L_0 is the complement of the union of some set of k parts in the random partition imply that

$$\mathrm{Inf}_f([n] \setminus S_B) \geq \mathrm{Inf}_f(L_0) \geq \frac{\epsilon^2}{400}$$

and, under the assumed accuracy of EstimateInf calls, the algorithm rejects f in Step 17. ◀

▶ **Claim 3.3** (Soundness II). *If f is $\frac{\epsilon}{100}$-close to a k-junta, but is $\frac{99\epsilon}{100}$-far from $\mathcal{F}$, then the* Implicit Learning Tester *rejects with probability at least $\frac{5}{6}$.*

Proof. Let g be any k-junta that satisfies $\mathrm{dist}_2(f, g) \leq \frac{\epsilon}{100}$. For any $h_{\mathrm{core}} \in \mathcal{F}_{\mathrm{core}}^{(\frac{\epsilon}{1000})}$ and any injective mapping $\psi : \{0,1\}^n \to \{0,1\}^k$, the function $h = h_{\mathrm{core}} \circ \psi$ is in $\mathcal{F}^{(\frac{\epsilon}{1000})}$ and so by the triangle inequality,

$$dist_2(f, \mathcal{F}^{(\frac{\epsilon}{1000})}) \geq dist_2(f, \mathcal{F}) - \tfrac{\epsilon}{1000}$$

and

$$\mathrm{dist}_2(g, h) \geq \mathrm{dist}_2(f, h) - \mathrm{dist}_2(f, g) \geq \tfrac{99}{100}\epsilon - \tfrac{\epsilon}{1000} - \tfrac{\epsilon}{100} \geq \tfrac{97}{100}\epsilon.$$

Then, by Proposition 2.8 and the union bound over all $|\mathcal{F}_{\mathrm{core}}^{(\frac{\epsilon}{1000})}| \leq (1000/\epsilon)^{2^k}$ functions in $\mathcal{F}_{\mathrm{core}}^{(\frac{\epsilon}{1000})}$, with probability at least $\frac{5}{6}$, the condition in Step 20 is never satisfied and the algorithm rejects. ◀

To complete the proof of Theorem 1.3 in the case where $p = 2$, consider now any property $\mathcal{P}$ that contains only functions which are $\frac{\epsilon}{10^6}$-close to some k-junta. Let $\mathcal{F}$ be the property that includes all k-juntas that are $\frac{\epsilon}{10^6}$-close to $\mathcal{P}$. Claim 3.1 shows that Implicit Learning Tester accepts every function in $\mathcal{P}$ with the desired probability, and Claims 3.2 and 3.3 shows that it rejects all functions that are ϵ-far from $\mathcal{P}$. Finally, we note that the query complexity of the algorithm is at most $q + 2m(2^{O(k\log(k))} + 2^k q) = \frac{2^{O(k\log k)}}{\epsilon^{10}}$, as claimed. Finally, the general result for ℓ_p testing when $p \neq 2$ follows from Fact 2.1.

3.2 Proof of Lemma 3

Let f be any function ε-close to a k-junta and assume without loss of generality (by relabeling the input variables if necessary) that f is close to a junta on $[k]$. The definition of $P_1, \ldots, P_{100k^4}$ in step 3, means that $S_{P_1}, \ldots, S_{P_{100k^4}}$ is a random partition of $[n]$. So by the union bound, the probability that any two of the coordinates in $[k]$ land in the same part is at most $\frac{1}{100k^2}$.

For each $\ell = 0, 1, 2, \ldots, r$, let $L_\ell = [n] \setminus \bigcup_{i=1}^{k} S_{P_{\ell,i}}$ denote the set of variables that have been "eliminated" after ℓ iterations of the loop. Then $[n] \setminus S_B = L_r$ and

$$\mathrm{Inf}_f([n] \setminus S_B) = \mathrm{Inf}_f(L_0) + \sum_{\ell=1}^{r} \Big(\mathrm{Inf}_f(L_\ell) - \mathrm{Inf}_f(L_{\ell-1})\Big). \tag{1}$$

We bound both terms on the right-hand side of the expression separately.

By Proposition 2.8, we have $\mathrm{Inf}_f([n] \setminus [k]) \leq \varepsilon^2$ and so by the monotonicity of influence there is a choice of $J \subseteq [k^2]$ of size $|J| \leq k$ for which $\mathrm{Inf}_f([n] \setminus S_J) \leq \varepsilon^2$. The guaranteed accuracy on EstimateInf then implies that

$$\mathrm{Inf}_f(L_0) \leq (1 + \tfrac{2}{100k^2})\varepsilon^2. \tag{2}$$

Define $\mathcal{E} = \{\ell \leq r : (L_\ell \setminus L_{\ell-1}) \cap [k] \neq \emptyset\}$ to be the set of rounds for which the algorithm eliminated at least one of the coordinates in $[k]$. By this definition, each $\ell \in [r] \setminus \mathcal{E}$ satisfies $(L_\ell \setminus L_{\ell-1}) \cap [k] = \emptyset$ and

$$\begin{aligned}\sum_{\ell \in [r]\setminus\mathcal{E}} \mathrm{Inf}_f(L_\ell) - \mathrm{Inf}_f(L_{\ell-1}) &= \sum_{\ell\in[r]\setminus\mathcal{E}} \sum_{T:\, T\cap L_\ell \neq \emptyset \wedge T \cap L_{\ell-1} = \emptyset} \hat{f}(T)^2 \\ &\leq \sum_{T \subseteq [n]\setminus[k]} \hat{f}(T)^2 \leq \mathrm{Inf}_f([n] \setminus [k]) \leq \varepsilon^2. \end{aligned} \tag{3}$$

For each $\ell \in \mathcal{E}$, define $X_\ell = \{\cup_{i=1}^k S_{P_{\ell,i,1-(z^*_\ell)_i}} : S_{P_{\ell,i,1-(z^*_\ell)_i}} \cap [k] \neq \emptyset\}$ to be the set of coordinates in the parts that contain a coordinate in $[k]$ that was eliminated in the ℓth iteration of the loop. Let also $Y_\ell = \{\cup_{i=1}^k S_{P_{\ell,i,(z^*_\ell)_i}} : S_{P_{\ell,i,1-(z^*_\ell)_i}} \cap [k] \neq \emptyset\}$ be the coordinates in the parts that were kept instead. Then the guaranteed accuracy of EstimateInf and the choice of z^*_ℓ implies that

$$\mathrm{Inf}_f(L_\ell) \leq \mathrm{Inf}_f\big((L_\ell \setminus X_\ell) \cup Y_\ell\big) + 2\,\tfrac{\varepsilon^2}{100k^2}$$

and, therefore,

$$\begin{aligned}\sum_{\ell\in\mathcal{E}} \mathrm{Inf}_f(L_\ell) - \mathrm{Inf}_f(L_{\ell-1}) &\leq \frac{2\varepsilon^2}{1000k} + \sum_{\ell\in\mathcal{E}} \mathrm{Inf}_f\big((L_\ell \setminus X_\ell) \cup Y_\ell\big) - \mathrm{Inf}_f(L_{\ell-1}) \\ &\leq \frac{2\varepsilon^2}{1000k} + \sum_{\ell\in\mathcal{E}} \sum_{T:\, T\cap(L_\ell\setminus X_\ell)\neq\emptyset\wedge T\cap L_{\ell-1}=\emptyset} \hat{f}(T)^2 \\ &\quad + \sum_{\ell\in\mathcal{E}} \sum_{T:\, T\cap Y_\ell\neq\emptyset\wedge T\cap L_{\ell-1}=\emptyset} \hat{f}(T)^2. \end{aligned} \tag{4}$$

As above, since $(L_\ell \setminus X_\ell) \cap [k] = \emptyset$,

$$\sum_{\ell\in\mathcal{E}} \sum_{T:\, T\cap(L_\ell\setminus X_\ell)\neq\emptyset\wedge T\cap L_{\ell-1}=\emptyset} \hat{f}(T)^2 \leq \sum_{T\subseteq[n]\setminus[k]} \hat{f}(T)^2 \leq \varepsilon^2. \tag{5}$$

It remains to bound the last sum on the right-hand side of (4). By splitting up the terms in this sum according to whether $|T| \leq k$ or not, we obtain

$$\sum_{T:T\cap Y_\ell\neq\emptyset\wedge T\cap L_{\ell-1}=\emptyset} \hat{f}(T)^2 \leq \sum_{|T|\leq k} \hat{f}(T)^2 \cdot \mathbf{1}[T \cap Y_\ell \neq \emptyset] + \sum_{|T|>k} \hat{f}(T)^2 \cdot \mathbf{1}[T \cap L_{\ell-1} = \emptyset].$$

Let $Z \subseteq [n] \setminus [k]$ denote the set of coordinates that occur in one of the the original parts $S_{P_1}, \ldots, S_{P_{100k^4}}$ that also contains one of the elements in $[k]$. Then $Y_\ell \subseteq Z$ and

$$\sum_{\ell\in\mathcal{E}} \sum_{|T|\leq k} \hat{f}(T)^2 \cdot \mathbf{1}[T\cap Y_\ell \neq \emptyset] \leq \sum_{\ell\in\mathcal{E}} \sum_{|T|\leq k} \hat{f}(T)^2 \cdot \mathbf{1}[T\cap Z \neq \emptyset] \leq k \cdot \sum_{|T|\leq k} \hat{f}(T)^2 \cdot \mathbf{1}[T\cap Z \neq \emptyset].$$

The probability, over the choice of $P_1, \ldots, P_{100k^4}$, that $T \cap Z \neq \emptyset$ is at most $|T|/100k^3$, so the expected value of the last expression (again over the choice of the initial partition) is bounded above by

$$\begin{aligned}\mathrm{E}\Big[\sum_{\ell\in\mathcal{E}}\sum_{|T|\le k}\hat{f}(T)^2\cdot\mathbf{1}[T\cap Y_\ell\neq\emptyset]\Big] &\le k\cdot\sum_{|T|\le k, T\setminus[k]\neq\emptyset}\hat{f}(T)^2\cdot\Big(\frac{k}{100k^3}\Big)\\ &\le \frac{1}{100k}\cdot\mathrm{Inf}_f([n]\setminus[k])\le\frac{\varepsilon^2}{100k}.\end{aligned}\tag{6}$$

Lastly, since $L_0 \subseteq L_{\ell-1}$ for each $\ell \geq 1$,

$$\sum_{|T|>k}\hat{f}(T)^2\cdot\mathbf{1}[T\cap L_{\ell-1}=\emptyset]\le\sum_{|T|>k}\hat{f}(T)^2\cdot\mathbf{1}[T\cap L_0=\emptyset].$$

A set T can be disjoint from L_0 only when its elements are contained in at most k of the parts of the initial random partition, which happens with probability at most $\frac{1}{100k^2}$ when $|T| > k$, so

$$\begin{aligned}\mathrm{E}\Big[\sum_{\ell\in\mathcal{E}}\sum_{|T|>k}\hat{f}(T)^2\cdot\mathbf{1}[T\cap L_{\ell-1}=\emptyset]\Big] &\le \mathrm{E}\Big[k\sum_{|T|>k}\hat{f}(T)^2\cdot\mathbf{1}[T\cap L_0=\emptyset]\Big]\\ &\le\frac{1}{100k}\sum_{|T|>k}\hat{f}(T)^2\le\frac{\varepsilon^2}{100k},\end{aligned}\tag{7}$$

where the last inequality uses the fact that $\sum_{|T|>k}\hat{f}(T)^2 \leq \mathrm{Inf}_f([n]\setminus[k])$.

Combining the inequalities (1)–(7), we obtain that the expected value of $\mathrm{Inf}_f([n]\setminus S_B)$ is bounded above by

$$\mathrm{E}\left[\mathrm{Inf}_f([n]\setminus S_B)\right]\le(1+\tfrac{2}{100k^2})\varepsilon^2+\tfrac{2\varepsilon^2}{100k}+(1+\tfrac{2}{100k})2\varepsilon^2\le 4\varepsilon^2.$$

Applying Markov's inequality and adding the probability that the junta variables are completely separated in the partition $P_1, \ldots, P_{100k^4}$ completes the proof of the lemma.

3.3 Proof of Lemma 4

For any $X = (x^{(1)}, \ldots, x^{(q)})$, let $\mathrm{dist}_X(f_1, f_2) = \left(\frac{1}{q}\sum_{i=1}^{q}\left(f_1(x^{(i)}) - f_2(x^{(i)})\right)^2\right)^{1/2}$ denote the empirical distance between f_1 and f_2 according to X. To prove the lemma, we want to show that $\mathrm{dist}_X(f, h)$ is within the specified bounds.

The function dist_X is a metric, so we can apply the triangle inequality to obtain

$$\mathrm{dist}_X(f,h)\le\mathrm{dist}_X(f,g)+\mathrm{dist}_X(g,h).$$

By Hoeffding's inequality, when $x^{(1)}, \ldots, x^{(q)}$ are drawn independently and uniformly at random, the upper bound

$$\mathrm{dist}_X(f,g)\le\mathrm{dist}_2(f,g)+\varepsilon\le 2\varepsilon$$

holds except with probability at most $e^{-16q\varepsilon^4}$.

We now want to show that $\mathrm{dist}_X(g, h)$ is also close to $\mathrm{dist}_2(g, h)$. This analysis is a bit more subtle, however, because the choice of samples $x^{(1)}, \ldots, x^{(q)}$ is *not* independent of h (as it affects what mapping ψ will be chosen by the algorithm). So before we can apply concentration inequalities, we must "decouple" X and h. To do so, we introduce a new

random process for generating X. Let $\lambda : [n] \to \{0,1\}^q$ be chosen uniformly at random. This function corresponds to a random partition of the set $[n]$ of coordinates into 2^q parts. Let $\pi : \{0,1\}^q \to \{0,1\}^q$ be a random permutation. Then the random variable X obtained by setting $x_j^{(i)} = \pi(\lambda(j))_i$ has the desired uniform distribution over sequences of q vectors in $\{0,1\}^n$.

This random process is designed so that the choice of ψ in the algorithm (and therefore also h) is *independent* of π; the only information about X used in determining it is the identity of the parts defined by λ, not what values the coordinates in each parts receive on the q queries. Then

$$\mathop{\mathrm{E}}_{X}[\mathrm{dist}_X(g,h)] = \mathop{\mathrm{E}}_{\lambda,r}[\mathop{\mathrm{E}}_{\pi}[\mathrm{dist}_X(g,h)]]$$

where r represents the internal randomness of the algorithm outside of that used to generate X. With probability at least $k^2/2^q$, the partition λ completely separates the indices in J. Fix such a partition λ. Define $J^* = J \cup \mathrm{supp}(\psi)$. Then $|J^*| \leq 2k$. Define $Y = (y^{(1)}, \ldots, y^{(q)})$ by setting $y^{(i)} = x^{(i)}_{J^*}$. Since $\mathrm{dist}_X(g,h)$ only depend on the coordinates in J^*, we can write it equivalently as $\mathrm{dist}_Y(g,h)$.

Let D denote the distribution on Y induced by π. The distribution D is close to but not equal to the uniform distribution U on $\{0,1\}^{q\times|J^*|}$, since D is equivalent to the distribution obtained by making drawing $(y_i^{(1)}, \ldots, y_i^{(q)})$ for each $i \in J^*$ without replacement from $\{0,1\}^q$. Then

$$\begin{aligned}
&\Pr_{Y\sim D}[|\mathrm{dist}_Y(g,h) - \mathop{\mathrm{E}}_{Y\sim U}\mathrm{dist}_Y(g,h)| \geq \varepsilon] \\
&\qquad\leq \mathrm{d_{TV}}(D,U) + \Pr_{Y\sim U}[|\mathrm{dist}_Y(g,h) - \mathop{\mathrm{E}}_{Y\sim U}\mathrm{dist}_Y(g,h)| \geq \varepsilon] \\
&\qquad\leq \tfrac{4k^2}{2^q} + e^{-16q\varepsilon^4}.
\end{aligned}$$

In the last inequality, the bound $\mathrm{d_{TV}}(D,U) \leq \frac{(2k)^2}{2^q}$ is by the standard total variation bound between sampling with and without replacement [19] and the other bound on the other term is by Hoeffding's inequality.

4 Applications

In this short section, we show how Theorems 1.1 and 1.2 both follow directly from Theorem 1.3 and the junta theorem of Feldman and Vondrák.

Proof of Theorem 1.1. By the first part of the Feldman–Vondrák junta theorem, every submodular function $f \in \mathcal{F}_n$ is $\frac{\epsilon}{10^6}$-close to a k-junta for some $k = O(\frac{1}{\epsilon^2}\log\frac{1}{\epsilon})$. Therefore, by Theorem 1.3, submodularity can be tested with $2^{O(k\log k)}/\epsilon^{10} = 2^{\tilde{O}(1/\epsilon^2)}$ queries in the ℓ_2 testing model. By Fact 2.1, the number of queries for testing submodularity in the ℓ_p testing model for any $1 \leq p < 2$ is also $2^{\tilde{O}(1/\epsilon^2)}$ and for any $p > 2$ it is $2^{\tilde{O}(1/(\epsilon^{p/2})^2)} = 2^{\tilde{O}(1/\epsilon^p)}$. ◀

Proof of Theorem 1.2. By Lemma 1, additive functions, coverage functions, unit demand functions, OXS functions, and gross substitute functions are all also submodular. Therefore, the first part of the Feldman–Vondrák junta theorem also applies to these functions and the rest of the proof is identical to that of Theorem 1.1.

Lemma 1 also implies that fractionally subadditive functions are self-bounding, so the second part of the Feldman–Vondrák junta theorem shows that every function f that has either of these properties is $\frac{\epsilon}{10^6}$-close to a k-junta for some $k = 2^{O(\frac{1}{\epsilon^2})}$. Therefore,

by Theorem 1.3, fractional subadditivity and self-boundedness can both be tested with $2^{O(k \log k)}/\epsilon^{10} = 2^{2^{\tilde{O}(1/\epsilon^2)}}$ queries in the ℓ_2 testing model; the general result for the ℓ_p testing model again follows directly from Fact 2.1. ◀

Acknowledgments. The authors wish to thank the anonymous referees for valuable feedback, Amit Levi for insightful discussions and Karl Wimmer for pointing out [31] to us.

References

1 Ashwinkumar Badanidiyuru, Shahar Dobzinski, Hu Fu, Robert Kleinberg, Noam Nisan, and Tim Roughgarden. Sketching valuation functions. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1025–1035. SIAM, 2012. URL: `http://portal.acm.org/citation.cfm?id=2095197&CFID=63838676&CFTOKEN=79617016`.

2 Maria-Florina Balcan, Florin Constantin, Satoru Iwata, and Lei Wang. Learning valuation functions. In Shie Mannor, Nathan Srebro, and Robert C. Williamson, editors, *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, volume 23 of *JMLR Proceedings*, pages 4.1–4.24. JMLR.org, 2012. URL: `http://www.jmlr.org/proceedings/papers/v23/balcan12b/balcan12b.pdf`.

3 Maria-Florina Balcan and Nicholas J. A. Harvey. Learning submodular functions. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 793–802, 2011. `doi:10.1145/1993636.1993741`.

4 Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. L_p-testing. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 164–173. ACM, 2014. `doi:10.1145/2591796.2591887`.

5 Eric Blais. Testing juntas nearly optimally. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 151–158. ACM, 2009. `doi:10.1145/1536414.1536437`.

6 Eric Blais, Amit Weinstein, and Yuichi Yoshida. Partially symmetric functions are efficiently isomorphism testable. *SIAM J. Comput.*, 44(2):411–432, 2015. `doi:10.1137/140971877`.

7 Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.*, 47(3):549–595, 1993. `doi:10.1016/0022-0000(93)90044-W`.

8 Deeparnab Chakrabarty and Zhiyi Huang. Recognizing coverage functions. *SIAM Journal on Discrete Mathematics*, 29(3):1585–1599, 2015. `arXiv:http://dx.doi.org/10.1137/140964072`, `doi:10.1137/140964072`.

9 Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Efficient sample extractors for juntas with applications. In Luca Aceto, Monika Henzinger, and Jirí Sgall, editors, *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part I*, volume 6755 of *Lecture Notes in Computer Science*, pages 545–556. Springer, 2011. `doi:10.1007/978-3-642-22006-7_46`.

10 Ilias Diakonikolas, Homin K. Lee, Kevin Matulef, Krzysztof Onak, Ronitt Rubinfeld, Rocco A. Servedio, and Andrew Wan. Testing for concise representations. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 549–558. IEEE Computer Society, 2007. `doi:10.1109/FOCS.2007.32`.

11 Ilias Diakonikolas, Homin K. Lee, Kevin Matulef, Rocco A. Servedio, and Andrew Wan. Efficiently testing sparse GF(2) polynomials. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata,*

Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I: Tack A: Algorithms, Automata, Complexity, and Games, volume 5125 of *Lecture Notes in Computer Science*, pages 502–514. Springer, 2008. doi:10.1007/978-3-540-70575-8_41.

12 Irit Dinur and Samuel Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162:2005, 2004.

13 Uriel Feige. On maximizing welfare when utility functions are subadditive. *SIAM J. Comput.*, 39(1):122–142, 2009. doi:10.1137/070680977.

14 Uriel Feige, Vahab S. Mirrokni, and Jan Vondrák. Maximizing non-monotone submodular functions. *SIAM J. Comput.*, 40(4):1133–1153, 2011. doi:10.1137/090779346.

15 Vitaly Feldman and Pravesh Kothari. Learning coverage functions and private release of marginals. In Maria-Florina Balcan, Vitaly Feldman, and Csaba Szepesvári, editors, *Proceedings of The 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13-15, 2014*, volume 35 of *JMLR Workshop and Conference Proceedings*, pages 679–702. JMLR.org, 2014. URL: http://jmlr.org/proceedings/papers/v35/feldman14a.html.

16 Vitaly Feldman and Jan Vondrák. Tight bounds on low-degree spectral concentration of submodular and XOS functions. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 923–942. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.61.

17 Vitaly Feldman and Jan Vondrák. Optimal bounds on approximation of submodular and XOS functions by juntas. *SIAM J. Comput.*, 45(3):1129–1170, 2016. doi:10.1137/140958207.

18 Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky. Testing juntas. *J. Comput. Syst. Sci.*, 68(4):753–787, 2004. doi:10.1016/j.jcss.2003.11.004.

19 David Freedman. A remark on the difference between sampling with and without replacement. *Journal of the American Statistical Association*, 72(359):681–681, 1977. doi:10.1080/01621459.1977.10480637.

20 Ehud Friedgut. On the measure of intersecting families, uniqueness and stability. *Combinatorica*, 28(5):503–528, 2008. doi:10.1007/s00493-008-2318-9.

21 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998. doi:10.1145/285055.285060.

22 Parikshit Gopalan, Ryan O'Donnell, Rocco A. Servedio, Amir Shpilka, and Karl Wimmer. Testing fourier dimensionality and sparsity. *SIAM J. Comput.*, 40(4):1075–1100, 2011. doi:10.1137/100785429.

23 Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. *SIAM J. Comput.*, 22(6):1331–1348, 1993. doi:10.1137/0222080.

24 Benny Lehmann, Daniel J. Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006. doi:10.1016/j.geb.2005.02.006.

25 Kevin Matulef, Ryan O'Donnell, Ronitt Rubinfeld, and Rocco A. Servedio. Testing halfspaces. *SIAM J. Comput.*, 39(5):2004–2047, 2010. doi:10.1137/070707890.

26 Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. URL: http://www.cambridge.org/de/academic/subjects/computer-science/algorithmics-complexity-computer-algebra-and-computational-g/analysis-boolean-functions.

27 Michal Parnas, Dana Ron, and Alex Samorodnitsky. Testing basic boolean formulae. *SIAM J. Discrete Math.*, 16(1):20–46, 2002. URL: http://epubs.siam.org/sam-bin/dbq/article/40744.

28 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996. doi:10.1137/S0097539793255151.

29 Rocco A. Servedio. Testing by implicit learning: A brief survey. In *Property Testing - Current Research and Surveys*, pages 197–210, 2010. doi:10.1007/978-3-642-16367-8_11.

30 C. Seshadhri and Jan Vondrák. Is submodularity testable? In Bernard Chazelle, editor, *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 7-9, 2011. Proceedings*, pages 195–210. Tsinghua University Press, 2011. URL: http://conference.itcs.tsinghua.edu.cn/ICS2011/content/papers/21.html.

31 Karl Wimmer and Yuichi Yoshida. Testing linear-invariant function isomorphism. In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, volume 7965 of *Lecture Notes in Computer Science*, pages 840–850. Springer, 2013. doi:10.1007/978-3-642-39206-1_71.

A Missing proofs from Section 2

We begin with the proof of Lemma 2. We emphasize that the proof below is essentially as found in [6]; the reason we include it here is that the original statement of the proof only applied to Boolean-valued functions. As we see below, however, the same argument also holds for real-valued functions.

▶ **Theorem A.1.** *(Dinur and Safra [12]; Friedgut [20]) Let $\mathcal{G}$ be a t-intersecting family of subsets of $[n]$ for some $t \geq 1$. For any $p < \frac{1}{t+1}$, the p-biased measure of $\mathcal{G}$ is bounded by $\mu_p(\mathcal{G}) \leq p^t$.*

Proof of Lemma 2. For $0 \leq t \leq \frac{1}{2}$, let $\mathcal{G}_t = \{J \subseteq [n] : \mathrm{Inf}_f(\overline{J}) < t\varepsilon^2\}$ be the family of all the sets whose compliments have influence less than $t\varepsilon^2$. For any two sets $J, K \in \mathcal{G}_{\frac{1}{2}}$, the subadditivity of influence implies that

$$\mathrm{Inf}_f(\overline{J \cap K}) = \mathrm{Inf}_f(\overline{J} \cup \overline{K}) \leq \mathrm{Inf}_f(\overline{J}) + \mathrm{Inf}_f(\overline{K}) < \varepsilon^2.$$

But f is ε-far from every k-junta, so for any two sets $J, K \in \mathcal{G}_{\frac{1}{2}}$, $|J \cap K| > k$, from Proposition 2.8. Which means $\mathcal{G}_{\frac{1}{2}}$ is a $k+1$ intersecting family. There are two cases now, first one is, there is at least one set $J \in \mathcal{G}_{\frac{1}{2}}$ such that $|J| < 2k$, second one is all the sets $J \in \mathcal{G}_{\frac{1}{2}}$ will have $|J| \geq 2k$. We will show that in both the cases our lemma holds. In the first case let $J \in \mathcal{G}_{\frac{1}{2}}$ be a set which has fewer than $2k$ elements, with high probability the set J is completely separated by the partition $\mathcal{P}$, and we know that for any $K \in \mathcal{G}_{\frac{1}{2}}$, $|J \cap K| \geq k+1$, which means K is not covered by any union of k-parts in $\mathcal{P}$. Therefore, $\mathrm{Inf}_f(\overline{J}) \geq \frac{\varepsilon^2}{2} > \frac{\varepsilon^2}{4}$ as we wanted to show.

Consider the case where, all the sets in $\mathcal{G}_{\frac{1}{2}}$ have more than $2k$ elements. Then $G_{\frac{1}{4}}$ is a $2k$ intersecting family. Otherwise, if there are two sets $J, K \in \mathcal{G}_{\frac{1}{4}}$ such that $|J \cap K| < 2k$, then $\mathrm{Inf}_f(\overline{J \cap K}) \leq \mathrm{Inf}_f(\overline{J}) + \mathrm{Inf}_f(\overline{K}) < \frac{\varepsilon^2}{4} + \frac{\varepsilon^2}{4} < \frac{\varepsilon^2}{2}$, thus contradicting our assumption.

Let $J \subseteq [n]$ be the union of k parts in $\mathcal{P}$. Since $\mathcal{P}$ is a random partition, J is a random subset obtained by including each element of $[n]$ in J independently with probability $p = \frac{k}{r} < \frac{1}{2k+1}$. By Theorem A.1, $\Pr_{\mathcal{P}}[I_f(\overline{J}) < \frac{\varepsilon^2}{4}] = \Pr[J \in \mathcal{G}_{\frac{1}{4}}] = \mu_{\frac{k}{r}}(\mathcal{G}_{\frac{1}{4}}) \leq (\frac{k}{r})^{2k}$. By the union bound the probability that there exists a set $J \subseteq [n]$ that is the union of k parts in $\mathcal{P}$ for which $\mathrm{Inf}_f(\overline{J}) < \frac{\varepsilon^2}{4}$ is bounded above by $\binom{r}{k}(\frac{k}{r})^{2k} \leq (\frac{er}{k})^k(\frac{k}{r})^{2k} \leq (\frac{ek}{r})^k < \frac{1}{6}$. ◀

The proof of Proposition 2.9 is obtained by considering the EstimateInf algorithm below.

Algorithm 2: EstimateInf(f, S, m)

1 Draw $x_1, \ldots, x_m$ uniformly and independently at random from $\{0,1\}^{\overline{S}}$;
2 Draw $y_1, \ldots, y_m, y'_1, \ldots, y'_m$ uniformly and independently at random from $\{0,1\}^{S}$;
3 Return $\frac{1}{2m}\sum_{i=1}^{m}\left(f(x_i, y_i) - f(x_i, y'_i)\right)^2$;

The concentration of the estimated influence is obtained via the following (standard) version of Hoeffding's inequality.

▶ **Hoeffding's inequality.** *Let $X_1, ..., X_n$ be independent random variables bounded by $a_1 \leq X_i \leq b_i$. Let $X = X_1 + X_2 + \cdots X_n$ have expected value $E[X] = \mu$. Then for any $t > 0$,*

$$\Pr[|X - \mu| \geq t] \leq 2e^{-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}}.$$

Algorithmic Aspects of Private Bayesian Persuasion*

Yakov Babichenko[1] and Siddharth Barman[2]

1 **The William Davidson Faculty of Industrial Engineering and Management, Technion - Israel Institute of Technology, Haifa, Israel**
yakovbab@tx.technion.ac.il

2 **Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India**
barman@csa.iisc.ernet.in

Abstract

We consider a multi-receivers Bayesian persuasion model where an informed sender tries to persuade a group of receivers to take a certain action. The state of nature is known to the sender, but it is unknown to the receivers. The sender is allowed to commit to a signaling policy where she sends a private signal to every receiver. This work studies the computation aspects of finding a signaling policy that maximizes the sender's revenue.

We show that if the sender's utility is a submodular function of the set of receivers that take the desired action, then we can efficiently find a signaling policy whose revenue is at least $(1 - 1/e)$ times the optimal. We also prove that approximating the sender's optimal revenue by a factor better than $(1 - 1/e)$ is NP-hard and, hence, the developed approximation guarantee is essentially tight. When the sender's utility is a function of the number of receivers that take the desired action (i.e., the utility function is anonymous), we show that an optimal signaling policy can be computed in polynomial time. Our results are based on an interesting connection between the Bayesian persuasion problem and the evaluation of the concave closure of a set function.

1998 ACM Subject Classification F.2.0 Analysis of Algorithms and Problem Complexity

Keywords and phrases Economics of Information, Bayesian Persuasion, Signaling, Concave Closure

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.34

1 Introduction

Sender-receiver models have been extensively studied in economics to understand the role of information in strategic settings; see, e.g., [5, 2, 12, 18]. Since these models study how information, or lack thereof, affects strategic decisions, they have found applications in various domains such as voting [2, 27], regulation policies [21, 28], marketing [8, 3], and auctions [7]. In these models, there is a sender who is more informed than the receiver, and the receiver has to take an action that affects both the sender's and the receiver's utility. An important objective behind studying these models is to quantify the *informational advantage* of the sender. In particular, the goal is to understand the optimal policy using which the sender can transmit (partial) information – or, equivalently send signals – to persuade the receiver into taking an action that is beneficial for the sender. Therefore, research work in

* The first author was supported by Israel Science Foundation, grant # 2021296. The second author was supported in part by Ramanujan Fellowship (SB/S2/RJN-128/2015).


© Yakov Babichenko and Siddharth Barman;
licensed under Creative Commons License CC-BY

this field is aimed at characterizing signaling policies that yield the sender the maximum possible revenue.

A fundamental sender-receiver model considered in the literature is the *Bayesian persuasion* model, wherein the sender is allowed to *commit* to a signaling (information-revelation) policy *before* she receives the information.[1] In this model, the utilities of the sender and the receiver (obtained from different sender/receiver actions) depend on the state of nature, which is drawn from a prior distribution. This distribution is known to the receiver and the sender. But, only the sender knows the realized state of nature (i.e., the draw) and, hence, has an informational advantage. In this Bayesian model – building upon the classical work by Aumann and Maschler [5] – Kamenica and Gentzkow [18] study the case in which there is exactly one receiver and they obtain a clean characterization of the sender's optimal signaling policy and her optimal revenue. However, this clean characterization fails to hold when there are multiple receivers; this is true even if the receivers have no payoff externalities.

In fact, except for a few specific instances [29, 11], little is known about optimal signaling policies in settings wherein the sender has to transmit to multiple receivers. To address this limitation, in a recent work Arieli and Babichenko [4] examine the following multiple-receiver setting, which will also be the focus of the our work: there are two states of nature, a single sender, and n receivers with binary ($\{0,1\}$) actions. Like the standard Bayesian setting, in this model the state of nature is drawn from a prior distribution that is known to the receivers and the sender; but, only the sender has access to the realized state. Each receiver's utility depends on her action and the state of nature (which she does not know). Furthermore, in this model the sender is allowed to send *private* signals to the receivers, and her utility is a monotonic set function of the *profile* of actions of the receivers, i.e., the sender's utility is a function which depends upon the set of receivers that, say, play action one.

Our goal is to examine the algorithmic aspects of this model. In particular, we focus on the computation of an (approximately) optimal policy for the sender. When the sender's utility is a submodular function, we show that a signaling policy whose revenue is at least $(1-\frac{1}{e}-\epsilon)$ times the sender's optimal revenue can be found in polynomial time; here $\epsilon > 0$ is an arbitrarily small constant. In addition, we establish that there does not exist a polynomial-time algorithm which approximates the sender's optimal revenue by a factor better than $(1-\frac{1}{e}+\epsilon)$, unless P = NP. Hence, the obtained approximation ratio is almost tight. For the case in which the sender's utility is anonymous – i.e., depends only on the number of receivers that play action one and not on their identities – we develop a polynomial algorithm for finding the optimal policy.

Our results are based on an interesting connection between the sender's optimal policy and the *concave closure* of a set function. We observe that computation/approximation, of an optimal signaling policy is equivalent to computation/approximation of the concave closure. Our signaling results follow from the following analogous results for the concave closure: a tight $(1-\frac{1}{e})$ approximation bound for the concave closure of submodular functions and a polynomial-time algorithm for the concave closure of anonymous functions. Since concave closure is a fundamental object of study in discrete convex analysis (see, e.g., [26]) our results for the evaluation of the concave closure might be of independent interest.

Although our focus is on a basic setting (in particular, on a setting in which the receivers have no payoff externalities,[2] they have binary actions, and the state of nature is binary), the considered model does capture several interesting – albeit stylized – scenarios.

---

[1] The Bayesian persuasion model complements the classical *cheap talk* model [12] which assumes that sender decides which signal to send after she receives the information.

[2] That is, each receiver's utility depends on her action and the state of nature, and not on the action of other receivers.

For example, the model can be used to represent a marketer (a sender) who is trying to persuade consumers (the receivers) to buy a certain product. Here, it is natural to assume that the marketer has more information about the quality of the product than the consumers. Also, different potential consumers may have different utilities for adopting the product. Note that each consumer has two possible (binary) actions, either to buy the product or not. In addition, we can consider the product (state of nature) to be in one of two possible states: the product is either high-quality or low-quality. Furthermore, the marketer's production cost may not be linear. Since it is reasonable to assume that the marketer's utility depends on the number of consumers that buy the product, this example is exactly captured by our result on anonymous utilities. In particular, using the algorithm developed for anonymous utilities we can efficiently compute an optimal signaling policy for the marketer.

Along these lines, the considered model also captures a viral marketing scenario where the receivers are the "opinion leaders", say, in a social network. After persuading a subset of "opinion leaders" to adopt a product, the information about the product will be spread through the network according to some diffusion process. As demonstrated in the notable work for Kempe et al. [23], many diffusion processes satisfy the submodularity property. Hence, our result on submodular utilities can be applied in such settings.

Another example is that of a lobbyist who is trying to persuade politicians to support a certain proposal. The proposal will pass if the number of supporters is above a specified threshold. Under the assumption that politicians vote sincerely[3] (i.e., they vote in favor of the alternative that maximizes their utility, given the information they posses about the proposal), this example is captured by the anonymous supermajority utility case.

Techniques. As mentioned above, our proofs are based on an interesting connection between the Bayesian persuasion model and the *concave closure of a set function* [25, 13, 30]. Specifically, in Section 2 (Lemma 1) we show that computation (or approximation) of sender's optimal revenue with utility $V : 2^{[n]} \to \mathbb{R}_+$ is (computationally) equivalent to evaluation (or approximation) of the concave closure of the function V, here $[n]$ is the set of receivers. Concave closure has been studied in the context of submodular maximization, see, e.g., [10] and [30]. In particular, prior work has shown that even though the concave closure provides a tight relaxation for constrained maximization of submodular functions, it is NP-hard to compute. Hence, instead of focusing on the computation of the concave closure, approximation results for submodular maximization typically rely on finding the *multilinear relaxation*. It turns out that in the context of the Bayesian persuasion problem the concave closure is not just a technical tool, but a core object. Therefore, a key focus of the paper is on efficiently approximating concave closures.

Specifically, we develop a $(1 - \frac{1}{e})$-approximation algorithm for computing the concave closure of monotone submodular functions. Our tight approximation result rests on an approximation preserving reduction between computing the concave closure and the problem of maximizing a monotone submodular function subject to a matroid constraint. We obtain such a reduction by a careful rounding of the problem parameter to a discrete grid. Since submodular maximization under matroid constraints admits a $(1 - \frac{1}{e})$ approximation (see [10]) the desired result follows. For the hardness result, we use tools from [24] and [17] which were developed to establish the hardness of approximating the maximum social welfare in combinatorial auctions.

[3] Note that this simplifying assumption assumes that politicians adopt a simple behavioral rule rather than a sophisticated equilibrium behavior. In equilibrium, each politician should condition his vote on the event that his vote will be pivotal, which leads the politicians to a completely different behavior, see e.g., [6].

We establish the result for anonymous utility functions (which are not necessarily submodular/concave) by developing a polynomial-time algorithm for computing the concave closure of anonymous functions. This result builds upon a lemma from [4], which characterizes the maximum "mass" that can be assigned to subsets of size $k \leq n$ under given marginal constraints; see Lemma 3 below for details. In addition to Lemma 3, to obtain the result we show a non-trivial property that in this case the maximal assignment is "monotonous across all ks," see details in Lemma 4. This additional property allows us to formulate the original concave closure problem as an LP with a *polynomial* number of variables.

1.1 Additional Related Work

The current literature on Bayesian persuasion starts with the result of [18] who – building upon the classical work by [5] – analyze the case of a single sender and a single receiver. Several extensions of this model – including ones that consider multiple receivers – have been studied in recent years, see, e.g., [1, 19, 20]. The setting wherein the sender is only allowed to send a public signal to the receivers is considered in [2, 27]. Furthermore, the complementary setting in which the sender is allowed to send private signals to the receivers has been studied in [4, 29, 31]. Our result is most closely related to the work of [4] where the optimal policy and the optimal revenue are characterized for supermodular utilities, utilities that are both anonymous and submodular, and also for supermajority utilities. In particular, we build upon the work of [4] with a computation perspective. We show that for *every* anonymous utility the optimal revenue can be computed in polynomial time. This result generalizes the claim in [4] where the same result (along with a closed-form expression for the sender's revenue) was established for anonymous utilities that are also submodular. We provide a tight approximation result for submodular utilities. Our inapproximability result for submodular functions (Theorem 9) indicates that it is unlikely that there exists a closed-form expression for the sender's revenue when her utility function is submodular.

A number of recent results in the computer science community have examined algorithmic questions surrounding the above mentioned models and signaling in general [15, 14, 16, 22, 9]. In particular, an interesting paper by Dughmi and Xu [15] studies the complexity of Bayesian persuasion in the single-receiver model of Kamenica and Gentzkow [18]. They consider the case in which the receiver has n actions and there are $\exp(n)$ states of nature. Dughmi and Xu [15] show that when the payoff profiles are i.i.d. distributed (for all receiver's actions) the problem can be solved in polynomial time. The same is not true if the payoff profiles are independently, but not identically, distributed – in this case the problem becomes #P-hard. Finally, they also show that the general problem (with arbitrary payoff profiles) can be approximately solved efficiently in a query model, if we assume that the receiver follows the recommended action by the sender in all cases where no ε-better action (according to her belief) exists.[4]

2 Notations and Preliminaries

We consider a Bayesian persuasion model with a sender and n receivers, $[n] = \{1, 2, ..., n\}$. Write $\Omega = \{\omega_0, \omega_1\}$ to denote the two possible states of nature. Each receiver $i \in [n]$ has two actions, $\{0, 1\}$, and a utility function, u_i, that depends on the state of nature and its own action, $u_i : \Omega \times \{0, 1\} \to \mathbb{R}$. All receivers share a common prior distribution, where

[4] This notion is called ε-incentive compatibility.

$0 < \gamma < 1$ is the probability of state ω_1, and $1-\gamma$ of state ω_0. Note that even though the receivers' utilities are dependent on the realized state of nature they are a priori unaware of it. Throughout, we will use $\Delta(A)$ to denote the set of probability distributions over set A.

It is shown in [4] that, without loss of generality, we can assume that $u_i(\omega_0, 0) > u_i(\omega_0, 1)$ and $u_i(\omega_1, 1) > u_i(\omega_1, 0)$ for all receivers $i \in [n]$. In particular, it is shown in [4] that we can efficiently reduce an instance with arbitrary utility functions to an instance in which the receivers prefer to play 1 when the state of nature is ω_1 and prefer to play 0 when the state is ω_0. Hence, throughout the paper we will work with this assumption on receivers' utilities.

As mentioned earlier, the sender's utility, V, depends on the set of receivers that play action 1, $V : \{0,1\}^n \to \mathbb{R}$. With a slight abuse of notation, for a subset $S \subseteq [n]$, we will use $V(S)$ to denote $V(1_S)$, where 1_S is the characteristic vector of subset S. Throughout we will assume that the sender's utility monotonically increases with the set of receivers that play action 1: $V(S) \leq V(T)$ for every $S \subseteq T$.

Note that we have restricted our attention to the case wherein the sender's utility does not depend on Ω. More generally, the sender's utility can be defined to be a function of the state of nature as well, i.e., we can have $V : \Omega \times \{0,1\}^n \to \mathbb{R}$. It is shown in [4] that such a general case can always be efficiently reduced to a setting in which $V(\omega_0, S) = V(\omega_1, S)$ for each S. Hence, in this paper we will focus on utility functions, V, that are state independent.

Recall that the utility function V is said to be submodular if it satisfies the decreasing marginal property. That is, for every subset $S \subset T \subset [n]$ and each $i \in [n] \setminus T$, we have $V(S \cup \{i\}) - V(S) \geq V(T \cup \{i\}) - V(T)$.

As is typical is Bayesian persuasion models, we assume that only the sender knows the realized state. The receivers remain unaware of it. Furthermore, following the model of Kamenica and Gentzkow [18] we allow the sender to commit in advance to an information-revelation/signaling policy. In this work, however, we allow the sender to reveal the information to every receiver privately. This translates to a state dependent signaling distribution. Formally a policy of the informed sender consists of n finite sets $\{\Theta_i\}_{i=1,\ldots,n}$, where Θ_i is the private signal set of receiver i, and a mapping $F : \Omega \to \Delta(\Theta_1 \times \cdots \times \Theta_n)$. Write $\Theta := \Theta_1 \times \cdots \times \Theta_n$. The sender can commit to a policy F that is known to the receivers prior to stage in which the state ω is realized.

The sequence of the interaction between the sender and the receivers is as follows. First, the sender commits to a signaling policy F. Then, a state $\omega \in \Omega$ is realized in accordance with the prior $(\gamma, 1-\gamma)$. After that a profile of signals $\theta = (\theta_1, \ldots, \theta_n)$ is generated according to the distribution $F(\omega)$. Every receiver i observes her private signal realization $\theta_i \in \Theta_i$ and forms a posterior $\mathbb{P}_F(\omega_1|\theta_i) = p(\theta_i)$.

With the posterior in hand, receiver i selects an action that maximizes her expected utility. In other worlds, receiver i plays action 1 if and only if

$$p(\theta_i)u_i(\omega_1, 1) + (1 - p(\theta_i))u_i(\omega_0, 1) \geq p(\theta_i)u_i(\omega_1, 0) + (1 - p(\theta_i))u_i(\omega_0, 0).$$

We assume that in case of indifference, receivers plays action 1. Let $g_i(\theta_i) \in \{0,1\}$ denote receiver i's best-reply action when she observes the signal θ_i. Also, write $g(\theta)$ to be the action profile of the receivers when the realized vector of signals is θ. We will use $F_1 \in \Delta(\Theta)$ to denote the signal distribution conditional on state ω_1 and $F_0 \in \Delta(\Theta)$ to denote the signal distribution conditional on state ω_0.

Let $s(F)$ be the sender's utility from the policy (Θ, F):

$$s(F) := \gamma \mathbb{E}_{\theta \sim F_1}[V(g(\theta))] + (1-\gamma)\mathbb{E}_{\theta \sim F_0}[V(g(\theta))]. \tag{1}$$

A signaling policy (Θ, F) is said to be *optimal* if it maximizes sender's utility among all possible signal sets Θ and all possible signals $F : \Omega \to \Delta(\Theta)$.

We begin by stating a result from [4] (see Lemma 1 in [4]) that shows the existence of an optimal policy with the following useful properties:

- For every receiver i, the private signal set Θ_i is equal to $\{0,1\}$ and receiver i's best reply $g_i(\theta_i) = \theta_i$. In other words, when signal θ_i is recommended by the sender to receiver i it is profitable (after receiver i performs a Bayesian update of her belief on the state of the world) for her to follow the recommendation. In [18], such policies are called *straightforward*.
- In the optimal policy $F_1(1,1,\ldots,1) = 1$, i.e., when state ω_1 is realized the sender recommends everyone to adopt the product. Recall that F_1 is a distribution over the set Θ, which (by the previous property) is $\{0,1\}^n$ for the optimal policy under consideration.
- When the realized state is ω_0, the sender recommends to receiver i to adopt the product with probability of at most $a_i := \min\left\{\frac{\gamma}{1-\gamma}\frac{u_i(\omega_1,1)-u_i(\omega_1,0)}{u_i(\omega_0,0)-u_i(\omega_0,1)}, 1\right\}$. Write marginal $F_0(\theta_i = 1) := \sum_{\theta\in\{0,1\}^n:\ \theta_i=1} F_0(\theta)$. We succinctly express this condition as $F_0(\theta_i = 1) \le a_i$. The number a_i can be interpreted as the maximal probability that the sender can "lie" to the receiver, and will be called the *persuasion level of player i*.

Under such an optimal policy, the sender's utility is given by

$$s(F) = \gamma V([n]) + (1-\gamma)\mathbb{E}_{\theta\sim F_0} V(\theta) \tag{2}$$

Overall, in light of these properties the problem of determining an optimal policy (over general signal sets Θ and mappings $F : \Omega \to \Delta(\Theta)$) reduces to the following well-structured maximization problem:

$$\text{maximize} \quad \mathbb{E}_{\theta\sim F_0} V(\theta) \quad \text{subject to} \quad F_0(\theta_i = 1) \le a_i \quad \forall i \in [n]. \tag{3}$$

Note that the prior γ and the utility $V([n])$ are fixed parameters. Hence, an optimal solution of (3) gives us an optimal solution of (2).

For each subset $S \subset [n]$, with characteristic vector 1_S, write μ_S to be the probability that exactly the receivers is S will receive the recommendation to adopt the product, i.e., $\mu_S := F_0(1_S)$. For a given persuasion levels profile $a := (a_1, ..., a_n)$, the maximization problem (3) can be written as

$$\begin{aligned} V^+(a) := \max \ & \sum_{S\subseteq[n]} \mu_S V(S) \\ \text{s.t.} \ & \sum_{S\subset[n]} \mu_S 1_S \le a \\ & \sum_{S\subset[n]} \mu_S = 1 \\ & \mu_S \ge 0. \end{aligned} \tag{4}$$

This is exactly the definition of the concave closure of the set function V evaluated at a given vector a.

Throughout, we will use $V^+(a)$ to denote the concave closure of the sender's utility function V at $a \in [0,1]^n$. We will refer to solving (approximating) the optimization problem (4) as computing (approximating) the concave closure. Interestingly, the concave closure has been studied in the optimization literature. In particular, it is considered as a technical tool for submodular maximization; see, e.g., [10]. Note that computing the concave closure corresponds to solving a linear programming with polynomial number of constraints, but with an exponential number of variables (the variables are μ_S for every $S \subset [n]$).

In some cases we will be interested in approximating the optimal revenue of the sender and, hence, we introduce here the following lemma that states that computing (approximating) the concave closure is computationally equivalent to computing (approximating) the sender's optimal revenue. Note that, for a given parameter $\alpha \in (0,1]$, an α approximation of the concave closure corresponds to a distribution $\{\mu_S\}_{S\subseteq[n]}$ that satisfies the feasibility constraints of the optimization problem (4) and obtains an objective function value, $\sum_{S\subseteq[n]} \mu_S V(S)$, that is at least α times the optimal. The next lemma states that there exists an approximation-preserving, polynomial-time reduction between computing the concave closure and finding the optimal revenue of the sender. Specifically, the lemma establishes that computing the concave closure of the sender's utility function lies at the core of determining a revenue-maximizing policy for the sender.

▶ **Lemma 1.** *Given a persuasion profile $a \in [0,1]^n$ and utility function V along with an α approximation of the concave closure $V^+(a)$. We can find, in polynomial time, a policy for the sender (with utility function V and persuasion profile a) that obtains revenue at least α times the optimal.*

Furthermore, for every $\varepsilon > 0$, there exists a polynomial-time reduction from the problem of α approximating a sender's revenue (with utility function V and persuasion profile a) to the problem of computing an $(\alpha+\varepsilon)$ approximation of the concave closure $V^+(a)$.

Proof. The forward direction is direct: by equation (2), an α approximation of $\max \mathbb{E}_{\theta\sim F_0} V(\theta)$ is also an α approximation of $\gamma V(N) + (1-\gamma)\mathbb{E}_{\theta\sim F_0} V(\theta)$.

For the other direction, given function V and persuasion level profile $a = (a_i)_{i\in[n]}$, we can set the prior γ to be very small (e.g., $\gamma = \frac{\varepsilon(V(N))^2}{1-\alpha}$ suffices) and we set receiver i utilities to be

$$u_i(\omega_0, 0) = 1,\ u_i(\omega_0, 1) = u_i(\omega_1, 0) = 0,\ u_i(\omega_1, 1) = a_i \frac{1-\gamma}{\gamma}.$$

Such a choice guarantees that indeed $a_i = \min\left\{\frac{\gamma}{1-\gamma}\frac{u_i(\omega_1,1)-u_i(\omega_0,1)}{u_i(\omega_0,0)-u_i(\omega_0,1)}, 1\right\}$. It follows that for such instances an α approximation of the sender's revenue implies $(\alpha+\varepsilon)$ approximation of the concave closure of V. ◀

In subsequent sections we establish algorithmic and hardness results for the problem of finding the optimal policy (and revenue) of the sender. We do so by using the above mentioned lemma and, in particular, addressing the computation of the concave closure.

3 Anonymous Utility

This section considers the case wherein the sender's utility function is anonymous i.e., it satisfies $V(S) = f(|S|)$ for some monotonically increasing function $f : [n] \to \mathbb{R}$. Our main result for anonymous utilities is as follows.

▶ **Theorem 2.** *There exists a polynomial algorithm for computing the maximum revenue and an optimal signaling policy for a sender that has a monotone, anonymous utility function.*

3.1 Proof of Theorem 2

We show that the concave closure of anonymous function can be computed in polynomial time.

We use $\mathcal{S}_k$ to denote all the size-k subsets of $[n]$, $\mathcal{S}_k := \{S \subseteq [n] \mid |S| = k\}$. We denote by $\mathsf{marg}(\mu)_i := \sum_{S \subset [n]: i \in S} \mu(S)$ the marginal probability of the ith coordinate to be equal 1. Note that the constraints of the concave closure $V^+(a)$ can be written as $\mathsf{marg}(\mu)_i \leq a_i$ for every $i \in [n]$.

The following lemma from [4] characterizes the maximum probability mass that can be assigned to subsets of size k under the constraints imposed by the profile $b = (b_1, ..., b_n)$.

▶ **Lemma 3** ([4]). *Let $1 \geq b_1 \geq b_2 \geq \ldots \geq b_n \geq 0$ be a monotonic sequence. The optimal value of the following maximization problem*

$$\begin{aligned} \max \quad & \sum_{S \in \mathcal{S}_k} \nu(S) \\ s.t. \quad & \sum_{S:\ i \in S} \nu(S) \leq b_i \qquad \forall i \in [n] \\ & \nu_S \geq 0 \end{aligned} \tag{5}$$

where ν is a positive measure (not necessarily a probability measure), is equal to

$$\beta_k(b_1, ..., b_n) = \min_{0 \leq m < k} \frac{1}{k-m}(b_{m+1} + \ldots + b_n).$$

Moreover, a measure ν that maximizes (5) *can be computed in polynomial time.*

The key idea is to use this lemma to solve the LP corresponding to the concave closure – which has exponential (in n) number of variables – by another LP that has a polynomial number of variables. First, we assume, without loss of generality, that the point a (where we want to evaluate the concave closure) satisfies $a_1 \geq a_2 \geq ... \geq a_n$. We split the original problem into n problems of finding a measure μ_k over $\mathcal{S}_k$ for every $k = 1, ..., n$ (the final measure is defined by $\mu = \mu_1 + ... + \mu_n$). The new maximization problem has n^2 variables $(a_i^j)_{i,j \in [n]}$, where $(a_1^j, ..., a_n^j)$ represents the marginal constrain vector on subsets of size k. We denote by α_k the measure that is assigned to subsets of size k, then the original maximization problem can be translated to the following

$$\begin{aligned} \max \ & \alpha_1 f(1) + \alpha_2 f(2) + ... + \alpha_n f(n) \\ \text{s.t.} \ & \sum_{k \in [n]} \alpha_k = 1,\ 0 \leq \alpha_k \leq \beta_k(a_1^k, a_2^k, ..., a_n^k) \text{ for } k \in [n], \text{ and } \sum_{j \in [n]} a_i^j \leq a_i. \end{aligned} \tag{6}$$

where the first constraint corresponds to $\sum_{S \subset [n]} \mu_S = 1$, the second follows from Lemma 3, and the last constraint uses the fact that marginals preserve additivity, and thus correspond to $\sum_{S \subset [n]} \mu_S 1_S \leq a$.

Note that the only nonlinear constraints in (6) are $\alpha_k \leq \beta_k(a_1^k, a_2^k, ..., a_n^k)$. Interesting, these constraints are "almost linear" in the following sense: *If* $a_1^k \geq a_2^k \geq ... \geq a_n^k$ then the constraint $\alpha_k \leq \beta_k(a_1^k, a_2^k, ..., a_n^k)$ can be written as

$$\begin{cases} \alpha_k \leq \frac{1}{k}(a_1 + a_2 + ... + a_n) \\ \alpha_k \leq \frac{1}{k-1}(a_2 + a_3 ... + a_n) \\ \vdots \\ \alpha_k \leq \frac{1}{1}(a_k + a_{k+1} + ... + a_n) \end{cases}$$

So the only obstacle is that the marginal constraint vector, in principle, in not guaranteed to satisfy the monotonicity constraint $a_1^k \geq a_2^k \geq ... \geq a_n^k$ (for all $k \in [n]$). The following Lemma 4 proves that, in fact, there always exists an optimal solution that satisfies this monotonicity

constraint (for all $k \in [n]$). Therefore we can impose this constraint in the optimization problem (6), and then it becomes an LP maximization with $poly(n)$ variables (and $poly(n)$ constraints).

We also note that the proof of Lemma 3 in [4] is constructive, and computationally efficient. Thus, to compute an optimal policy (not only optimal revenue) after we have computed the values of $(a_i^j)_{i,j\in[n]}$ that maximize (6) we can use the constrictive algorithm of the proof of Lemma 3 for all $k \in [n]$. This completes the proof of Theorem 2.

▶ **Lemma 4.** *Assume that the point a satisfies $a_1 \geq a_2 \geq ... \geq a_n$. For $(\mu_S)_{S\subset[n]}$ we denote by $a_i^k = \sum_{S\in\mathcal{S}_k:\ i\in S} \mu(S)$. There exists μ that maximizes* (6) *and satisfies in addition $a_1^k \geq a_2^k \geq ... \geq a_n^k$ for all $k \in [n]$.*

Proof. The proof builds upon ideas that were used in the proof of Lemma 3 in [4].

Let ν be a distribution that satisfies the constraints $\mathsf{marg}(\nu)_i \leq a_i$, and let $\alpha_k = \nu(\mathcal{S}_k)$ be the weight of ν on subsets of size k. It is sufficient to construct another distribution μ that satisfies $\mu(\mathcal{S}_k) = \alpha_k$ (and thus μ has the same revenue as ν), and in addition $a_1^k \geq a_2^k \geq ... \geq a_n^k$.

The construction is done in n steps, where the steps $k = n, n-1, ..., 1$ are done in an decreasing order. At step k we assign a measure of α_k to subsets of size k, and we denote the assigned measure by μ_k. Each step k is done in finite number of stages. Here we describe the assignment of measure at stage $k.m$.

During the construction we "assign mass" and thus, we "spend marginal constraints." We take track of the remaining marginal constraints vector. At the beginning, we set the constraints vector $(a_1^{n.0}, ..., a_n^{n.0}) = (a_1, ..., a_n)$ to be the original constraints.

During the process we preserve the monotonicity of the marginal constraints vector and therefore we can denote the marginal constraints vector at stage $k.m$ by

$$(a_1^{k.m}, ..., a_n^{k.m}) = (b_1, ..., b_j, \underbrace{c, c, ..., c}_{l-j \text{ times}}, b_{l+1}, ..., b_n)$$

where $b_j > c > b_{l+1}$ and $j < k \leq l$. Note that if $a_k^{k.j} = a_{k+1}^{k.j} = ... = a_n^{k.j}$ then $l = n$ and for simplicity of notation we denote $b_{n+1} = 0$. Note that if $a_1^{k.j} = a_2^{k.j} = ... = a_k^{k.j}$ then $j = 0$, and for simplicity of notation we denote $b_0 > b_1$.

At stage $k.m$, the idea is to distribute mass equally over the subsets S of size k that satisfy $[j] \subseteq S \subseteq [l]$ (we have $\binom{l-j}{k-j}$ such sets). If we do so, after we have distributed x units of mass the remaining marginal constraints vector will be

$$b(x) = (b_1 - x, ..., b_j - x, c - \frac{k-j}{l-j}x, ..., c - \frac{k-j}{l-j}x, b_{l+1}, ..., b_n) \tag{7}$$

because every element $i = j+1, j+2, ..., l$ appears in exactly $\frac{k-j}{l-j}$ fraction of the above subsets. Step $k.m$ terminates at the moment when one of the following three happens:

(1) The total mass that has been assigned during step k reaches α_k. In such a case we proceed to step $k-1$.

(2) The jth coordinate becomes equal to the $(j+1)$th coordinate. In such a case we proceed to stage $k.(m+1)$.

(3) The lth coordinate becomes equal to the $(l+1)$th coordinate. In such a case we proceed to stage $k.(m+1)$.

We denote by $\alpha_{k.m}$ the amount of mass that has been assigned during step $k.m$. We denote by $b(\alpha_{k.m})$ the marginal constraints vector after step $k.m$, where $b(\cdot)$ is defined in

equation (7). This marginal constraints serves as the marginal constraint vector for the next step (in case (1) happens) or the next stage (in case (2) or (3) happens).

We argue the following two statements, which will complete the proof.

1. The described process succeeds to complete all the n steps.
2. The described process at each step k assigns mass in a way that $a_1^k \geq a_2^k \geq ... \geq a_n^k$.

Statement (2) follows from the fact that at each stage $k.m$ the marginals of the assigned mass is of the form $(\underbrace{x,...,x}_{j_m \text{ times}}, \underbrace{cx,...,cx}_{l_m - j_m \text{ times}}, 0, ..., 0)$ for $x = \alpha_{k.m}$ and $c < 1$. Moreover, during step k the coordinate j_m is monotonically decreasing, and the coordinate l_m is monotonically increasing. Therefore, the sum of those vectors, which is equal to the vector $(a_1^k, a_2^k, ..., a_n^k)$ is monotonically increasing.

Assume by way of contradiction that statement (1) is false. The above process cannot assign the required measure only if we are at step k and the marginal constraints vector becomes $(d_1, d_2, ..., d_m, 0, ..., 0)$ for $m < k$. In such a case indeed the process cannot proceed, because it will turn the $m+1$ coordinate of the marginal constraint vector negative. We denote by α'_k the measure at step k that has been assigned up to the moment of termination.

We argue that this is impossible from the fact that $\alpha_n, \alpha_{n-1}, ..., \alpha_k$ are feasible weights for some distribution ν. The idea is that the described above process has minimal marginals on the elements $m+1, ..., n$, thus if this process cannot proceed neither could some other distribution ν. Formally, we denote $\nu = \nu_1 + ... + \nu_n$, where ν_j is a measure over $\mathcal{S}_j$. Note that $|\nu_j| = \alpha_j$. We denote $(d_i^j)_{i \in [n]}$ the marginals of ν_j. We argue that $\sum_{i=m+1}^n d_i^j \geq (j-m)\alpha_j$, because every subset of size j contains at least $j-m$ elements from the set $\{m+1, ..., n\}$. Therefore we have

$$a_{m+1} + ... + a_n \geq \sum_{j=k}^{n} \sum_{i=m+1}^{n} d_i^j \geq \sum_{j=k}^{n} (j-m)\alpha_j \tag{8}$$

On the other hand, the constructed measure μ_n with marginals $(a_i^j)_{i \in [n]}$ satisfies $\sum_{i=m+1}^n a_i^j = (j-m)\alpha_j$, because this process assigns positive probability *only* to subsets that contain $\{1, ..., m\}$ (because $m < k \leq j$ and $a_m^j > a_{m+1}^j$). Since the process spent all the marginal constraints $a_{m+1}, ..., a_n$ we have

$$a_{m+1} + ... + a_n = \sum_{j=k}^{n} \sum_{i=m+1}^{n} a_i^j = (k-m)\alpha'_k + \sum_{j=k+1}^{n} (j-m)\alpha_j < \sum_{j=k}^{n} (j-m)\alpha_j \tag{9}$$

Inequalities (8) and (9) yield a contradiction. ◀

4 Submodular Utilities

This section considers private Bayesian persuasion settings in which the sender's utility function is submodular. In particular, we develop a tight $(1 - 1/e)$ approximation of the optimal signaling policy when the sender's utility is a monotone submodular function.

It is relevant to note that our algorithmic results require only query access to the submodular function, i.e., our results hold as long as we can access to $V(S)$, for any subset $S \subseteq [n]$. This, in particular, implies that we can address submodular functions that admit a *succinct* representation.

We begin by noting that finding the concave closure of a submodular function is NP-hard: Given a succinct, monotone, submodular function $f : 2^{[n]} \to \mathbb{R}$ and a vector $a \in [0,1]^n$, it is NP-hard to compute the concave closure $f^+(a)$; see, e.g. [30] and [13].

It is relevant to note that while the concave closure of submodular functions are known to be computationally hard, approximation algorithms and inapproximability results for them have not been directly addressed in prior work.

4.1 Approximation Algorithm for Submodular Utilities

This section provides a $(1-1/e)$-approximation algorithm for computing the concave closure of a monotone, submodular function V. We obtain the $\left(1-\frac{1}{e}\right)$ approximation by reducing the computation of the concave closure to the problem of maximizing a submodular function subject to a matroid constraint. The key implication of this approximation result is the following theorem.

▶ **Theorem 5.** *If in a private Bayesian persuasion problem the utility of the sender, V, is a monotone, submodular function. Then, in polynomial time, we can compute a signaling policy that achieves a revenue of at least $(1-1/e-\varepsilon)$ times the optimal; here, ε is an arbitrarily small constant.*

4.1.1 Proof of Theorem 5

We show that the concave closure of submodular function V at any given vector $a=(a_1,a_2,\ldots,a_n)\in[0,1]^n$ can be approximated to within a factor of $\left(1-\frac{1}{e}-\varepsilon\right)$, for an arbitrarily small $\varepsilon>0$, then Theorem 5 follows from Lemma 1.

We split the marginal values a_i into two sets: $\{a_i : a_i \geq \frac{1}{n^2}\}$ are the *high values* and $\{a_i : a_i < \frac{1}{n^2}\}$ are the *low values*. Without loss of generality we assume that $a_1,...,a_m$ are the high values and $a_{m+1},...,a_n$ are the low values, for $m\leq n$.

Every distribution μ over subsets of $[n]$ induces a distribution $\nu=\nu(\mu)$ over subsets of $[m]$ in the following natural way: the probability mass μ_S on $S\subset[n]$ is moved to the set $S\cap[m]$, formally for each subset $T\subseteq[m]$ define $\nu_T:=\sum_{S\subset[n]:S\cap[m]=T}\mu_S$. The following lemma holds for distribution ν.

▶ **Lemma 6.** *For every distribution μ that satisfies the marginal constraints (i.e., $\sum_{S\subseteq[n]:S\ni i}\mu_S\leq a_i$) we have*

$$\sum_{S\subset[n]}\mu_S V(S)\leq\sum_{T\subset[m]}\nu_T V(T)+\sum_{i=m+1}^{n}a_i V(\{i\}).$$

Here $a_1,\ldots,a_m\geq\frac{1}{n^2}$ and $a_m,a_{m+1},\ldots,a_n\leq\frac{1}{n^2}$.

Proof.

$$\begin{aligned}\sum_{S\subset[n]}\mu_S V(S)&\leq\sum_{S\subset[n]}\mu_S\left[V(S\cap[m])+\sum_{i\in S,i>m}V(\{i\})\right]\\&=\sum_{T\subset[m]}\nu_T V(T)+\sum_{S}\sum_{i\in S,i>m}\mu_S V(\{i\})\\&=\sum_{T\subset[m]}\nu_T V(T)+\sum_{i>m}\sum_{S:i\in S}\mu_S V(\{i\})\\&\leq\sum_{T\subset[m]}\nu_T V(T)+\sum_{i>m}a_i V(\{i\}).\end{aligned}$$

Here, the first inequality follows from subadditivity of V. The second equation follows from the definition of $\nu = \nu(\mu)$. The third equation is obtained by changing the order of summation and the last inequality follows from the fact that μ satisfies the marginal constraints. ◀

We can consider the optimization problem corresponding to the concave closure restricted to the set $[m]$:

$$\begin{aligned} V_m^+(a) := \max & \sum_{T\subset[m]} \nu_T V(T) \\ \text{s.t.} & \sum_{T\subseteq[m]:T\ni i} \nu_T \le a_i \qquad \forall i \in [m] \\ & \nu \text{ is a probability measure.} \end{aligned} \tag{10}$$

Given a distribution $\overline{\nu}$ that α-approximates problem (10), we define distribution $\overline{\mu}$ over $[n]$ as follows: For each subset $T \subseteq [m]$, set $\overline{\mu}_T := (1-\frac{1}{n})\overline{\nu}$. In addition, for every $i > m$ set $\overline{\mu}_{\{i\}} := a_i$. Finally, to ensure that $\overline{\mu}$ is a probability measure we assign a probability mass of $c = \frac{1}{n} - a_{m+1} - ... - a_n > 0$ to the empty set, i.e., $\mu_\phi := c$.

▶ **Lemma 7.** *If distribution $\overline{\nu}$ α-approximates problem* (10)*, then $\overline{\mu}$ provides a $(1-\frac{1}{n})\alpha$-approximation of the original concave closure problem* (4)*.*

Proof. Recall that $V^+(a)$ denotes the concave closure of function V evaluated at a and, similarly, $V_m^+(a)$ is optimal value of (10).

$$\begin{aligned} \sum_{S\subset[n]} \overline{\mu}_S V(S) &= (1-\frac{1}{n}) \sum_{T\subset[m]} \overline{\nu}_T V(T) + \sum_{i>m} a_i V(\{i\}) \\ &\ge (1-\frac{1}{n})\alpha V_m^+(a) + \sum_{i>m} a_i V(\{i\}) \\ &\ge (1-\frac{1}{n})\alpha [V_m^+(a) + \sum_{i>m} a_i V(\{i\})] \\ &\ge (1-\frac{1}{n})\alpha V^+(a). \end{aligned}$$

Here the first equation is implied by the definition of $\overline{\mu}$. The second inequality follows from the fact that $\overline{\nu}$ α-approximates the concave closure (on the set $[m]$). The third inequality is trivial and the last one follows from Lemma 6. ◀

Lemma 7 reduces the original concave closure problem to the problem of computing the concave closure over $[m]$ where (unlike the original problem) we know that $a_i \ge \frac{1}{n^2}$ for each $i \in [m]$. In the remainder of the proof, we consider the later problem. The idea is to translate this problem into a discrete one. A natural way do to so is by rounding the underlying terms to integer multiples of a parameter $\delta := \frac{1}{n^4(n+1)}$ and then working with the multiples, instead of the fractional terms.

Since (10) is a linear program (over variables $\{\nu_T\}_{T\subseteq[m]}$) with at most $n+1$ non-trivial constraints, without loss of generality we can restrict attention to solutions that have support size of at most $n+1$.

As mentioned previously, we set a grid of size $\delta := \frac{1}{n^4(n+1)}$, and we consider the maximization problem of V_m^+ where we restrict the probabilities $\{\nu_S\}$ to be integer multiples of δ.

$$\begin{aligned} \max \quad & \sum_{T\subset[m]} \nu_T V(T) \\ \text{s.t.} \quad & \sum_{T\subseteq[m]:T\ni i} \nu_T \le a_i \qquad \forall i\in[m] \\ & \nu \text{ is a probability measure.} \\ & \nu_T \in \{0,\delta,2\delta,\dots,1\}. \end{aligned} \tag{11}$$

▶ **Lemma 8.** *If distribution $\widehat{\nu}$ be an α-approximate solution of optimization problem* (11) *with support size at most $n+1$. Then, $\widehat{\nu}$ is a $(1-\frac{1}{n\alpha})\alpha$-approximate solution of the concave closure $V_m^+(a)$ as well.*

Proof. We prove that if we restrict our attention to probabilities in the set $\{0,\delta,2\delta,\dots,1\}$, then we incur at most a multiplicative loss of $(1-\frac{1}{n\alpha})$. Given a distribution ν with support size at most $n+1$ we round down the probabilities to integer multiples of δ (and put all the remaining probability mass on the empty set), we denote the resulting distribution by ν'. Formally $\nu'_T=\ell\delta$ where $k=\max\{j\in\mathbb{Z}: j\delta\le\nu_T\}$. Note that

$$\begin{aligned} &\sum_T \nu_T V(T) - \sum_T \nu'_T V(T) = \sum_T (\nu_T-\nu'_T)V(T) \le \sum_T \delta V(T) \\ &\le \sum_{T\in\mathsf{supp}(\nu)} \frac{1}{n^4(n+1)} V([m]) \le \frac{1}{n^4} V([m]) \\ &\le \frac{1}{n^4}\sum_{i\in[m]} V(\{i\}) \le \frac{1}{n^2}\sum_{i\in[m]} a_i V(\{i\}), \end{aligned} \tag{12}$$

where the first equality is trivial. The first and the second inequality is a consequence of the rounding and the value of δ. The third inequality follows from the fact that the support size is at most $n+1$. The subadditivity of V gives us the fourth inequality and the last inequality follows from the fact that $a_i\ge\frac{1}{n^2}$.

Note also that $V_m^+(a)\ge\frac{1}{n}\sum_{i\in[m]} a_i V(\{i\})$ because one feasible option is to put a mass of $\frac{a_i}{n}$ on the singleton $\{i\}$, and the remaining probability mass to put on the empty set. This is indeed feasible because $\sum_i \frac{a_i}{n}\le\sum_i\frac{1}{n}\le 1$.

Finally let $\overline{\nu}$ be an α approximation for $V_m^+(a)$, and let $\overline{\nu}'$ be the corresponding rounding. Then

$$\begin{aligned} \frac{\sum_T \overline{\nu}'_T V(T)}{\sum_T \overline{\nu}_T V(T)} &= 1 - \frac{\sum_T \overline{\nu}_T V(T) - \sum_T \overline{\nu}'_T V(T)}{\sum_T \overline{\nu}_T V(T)} \\ &\ge 1 - \frac{\frac{1}{n^2}\sum_{i\in[m]} a_i V(\{i\})}{\sum_T \overline{\nu}_T V(T)} \\ &\ge 1 - \frac{\frac{1}{n^2}\sum_{i\in[m]} a_i V(\{i\})}{\alpha\frac{1}{n}\sum_{i\in[m]} a_i V(\{i\})} = 1-\frac{1}{n\alpha} \end{aligned}$$

where the first inequality follows from (12), and the second one from the fact that $V_m^+(a)\ge\frac{1}{n}\sum_{i\in[m]} a_i V(\{i\})$. ◀

By Lemma 8 we can restrict attention to the discrete problem (11). Note that the the discrete problem (11) is equivalent to

$$\begin{aligned} \max_{S^1,\dots,S^k\subseteq[m]} \quad & \frac{1}{k}\sum_{j=1}^{k} V(S^j) \\ \text{subject to} \quad & |\{j\in[k]\mid i\in S^j\}| \le k_i \qquad \forall i\in[n] \end{aligned} \tag{13}$$

where $k = n^4(n+1)$ and $k_i = a_i n^4(n+1)$ for all $i \in [n]$.

We complete the proof of the theorem by showing that (13) admits a $(1-1/e)$ approximation. We do so by showing that (13) corresponds to the problem of maximizing a monotone submodular function subject to a matroid constraint.

Consider base set $U = [m] \times [k]$. We get that the size of U is polynomially bounded. For a subset $R = \{(i_1, j_1), (i_2, j_2), \ldots, (i_l, j_l)\}$ of U and $j \in [k]$, write R^j to denote the projected subset $\{i' \in [m] \mid (i', j) \in R\}$.

With this notation in hand, define function F for each subset $R = \{(i_1, j_1), (i_2, j_2), \ldots, (i_l, j_l)\} \subset U$ as follows

$$F(R) := \frac{1}{k} \sum_{j=1}^{k} V(R^j). \tag{14}$$

We claim that F is submodular: consider subsets $X \subset Y \subset U$ and element $(i,j) \in U$. Note that $F(X + (i,j)) - F(X) = \frac{1}{k} V(X^j + i) - \frac{1}{k} V(X^j)$ and $F(Y + (i,j)) - F(Y) = \frac{1}{k} V(Y^j + i) - \frac{1}{k} V(Y^j)$. Since $X^j \subset Y^j$, the submodulartiy (monotonicity) of V implies the submodularity (monotonicity) of F.

Next we consider a partition matroid $\mathcal{M}$ over U. Specifically, we say that a subset $R \subset U$ is independent (with respect to the matroid $\mathcal{M}$) iff $|\{(i', j') \in R \mid i' = i\}| \leq k_i$ for all $i \in [n]$. Note that this is a partition matroid where the disjoint partitions are $B_i := \{(i,1), (i,2), \ldots, (i,k)\}$ and the cardinality bounds are k_is. In other words, we obtain $\mathcal{M}$ by defining R to be an independent subset iff $|R \cap B_i| \leq k_i$ for all i.

Note that if a subset $R \subset U$ is independent then $R^1, R^2, \ldots R^k$ satisfy the constraints of the optimization problem (13), i.e., for an independent R we have $|\{j \in [k] \mid i \in S^j\}| \leq k_i$ for all i.

Overall, we get that optimization problem (13) is equivalent to the following problem:

$$\begin{aligned} \max_{R \subset U} \quad & F(R) \\ \text{subject to} \quad & R \in \mathcal{M} \end{aligned}$$

Since this is a submodular maximization problem subject to a matroid constraint, it admits a $(1 - \frac{1}{e})$ approximation; see [10]. This in turn implies that the original problem admits a $(1 - \frac{1}{n})(1 - \frac{1}{0.62n})(1 - \frac{1}{e}) = (1 - \frac{1}{e} - O(\frac{1}{n}))$ approximation. We can set parameters such that instead of a multiplicative factor of $(1 - \frac{1}{n})(1 - \frac{1}{0.62n})$ in the approximation we get a term that is arbitrarily close one. Hence, we get the desired result.

4.2 Hardness of Approximating the Concave Closure

This section shows that the $\left(1 - \frac{1}{e}\right)$ approximation guarantee obtained in Section 4.1.1 is tight. In particular, applying the machinery developed by [24] leads us to the following theorem. We note that [24] establish the hardness of approximating maximum social welfare in combinatorial auctions and similar tools were developed in [17] for studying the inapproximability of the domatic number.

▶ **Theorem 9.** *Given a monotone, submodular fucntion $V : 2^{[n]} \to \mathbb{R}_+$ and vector $a \in [0,1]^n$, for any $\varepsilon > 0$, it is* NP*-hard to approximate the concave closure, $V^+(a)$, by a factor better than $\left(1 - \frac{1}{e} - \varepsilon\right)$.*

Proof Sketch. [24] study the combinatorial auction problem where n goods have to be partitioned among m agents whose utilities are submodular functions of the goods assigned

to them. In this problem, the objective is to maximize social welfare, i.e., the sum of the utilities of the receivers. It is shown in [24] that for this problem and any $\varepsilon > 0$ there does not exist a polynomial time algorithm that obtains an approximation ratio better than $\left(1 - \frac{1}{e} - \varepsilon\right)$, unless P = NP.

Specifically, [24] start with a *label-cover* problem where it is NP-hard to distinguish whether the optimal value, $OPT(L)$, is one or less than a particular constant, $c < 1$. From the given label cover problem they construct a combinatorial auction instance, I, wherein the maximum social welfare, $OPT(I)$ is greater than a threshold, τ if the label cover problem admits a solution of value one. Furthermore, if the optimal value of the label cover problem is less than c – i.e., $OPT(L) \leq c$ – then it must be the case that $OPT(I) \leq \left(1 - \frac{1}{e} - \varepsilon\right)\tau$. This, overall, establishes a $\left(1 - \frac{1}{e} - \varepsilon\right)$ hardness-of-approximation bound for the combinatorial auction problem.

Interestingly, in the constructed instance I all of the m receivers have the same monotone, submodular utility function, say, $f : 2^{[n]} \to \mathbb{R}_+$. We claim that approximating the concave closure of constructed function f at marginal vector $a := (\frac{1}{m}, \frac{1}{m}, \ldots, \frac{1}{m})$ by a factor better than $\left(1 - \frac{1}{e} - \varepsilon\right)$ is NP-hard. In particular, if $OPT(L) = 1$ then $f^+(a) \geq \tau'$ and, moreover, if $OPT(L) \leq c$ then $f^+(a) \leq \left(1 - \frac{1}{e} - \varepsilon\right)\tau'$; here τ' is a fixed parameter.

The proof of this claim can be obtained by considering the subsets in the support of an optimal solution, μ^*, of problem (4) defined for function f. Note that the proof given in [24] proceeds by considering the subsets that constitute the partition of goods among receivers in I, instead we can focus on subsets in the support of μ^* to obtain the result for the concave closure. In particular, the arguments presented in [24] go through if, instead of cardinalities, we consider measure of sets and expected values of quantities with respected to μ^*.[5] This, overall, establishes the desired inapproximability result for the concave closure. ◀

Acknowledgements. The authors thank Uriel Feige for helpful discussions and references.

References

1. Ricardo Alonso and Odilon Câmara. On the value of persuasion by experts. Technical report, University of Southern California, Marshall School of Business, 2014.
2. Ricardo Alonso and Odilon Câmara. Persuading voters. *Available at SSRN 2688969*, 2015.
3. Simon P Anderson and Régis Renault. The advertising mix for a search good. *Management Science*, 59(1):69–83, 2013.
4. Itai Arieli and Yakov Babichenko. Private bayesian persuasion. *Available at SSRN 2721307*, 2016.
5. Robert J Aumann, Michael Maschler, and Richard E Stearns. *Repeated games with incomplete information*. MIT press, 1995.
6. David Austen-Smith and Jeffrey S Banks. Information aggregation, rationality, and the condorcet jury theorem. *American political science review*, 90(01):34–45, 1996.
7. Dirk Bergemann and Martin Pesendorfer. Information structures in optimal auctions. *Journal of economic theory*, 137(1):580–609, 2007.

[5] For example, in Lemma 5 in [24], we can redefine sets N_1^e (N_2^e) to be the collection of subsets – instead of collection of players – in the support of μ^* that cover (do not cover) an edge e in the label cover instance. Along these lines, instead of bounding the cardinalities of N_1^e and N_2^e (which are denoted by n_1^e and n_2^e in [24]), we can bound the measures $\sum_{S \in N_1^e} \mu_S^*$ and $\sum_{S \in N_2^e} \mu_S^*$. Among other things, these changes allow us to use Jensen's inequality and bound Δ_e as specified in Lemma 5 by [24].

8 Fernando Branco, Monic Sun, and J Miguel Villas-Boas. Too much information? information provision and search costs. *Marketing Science*, 2015.

9 Peter Bro Miltersen and Or Sheffet. Send mixed signals: earn more, work less. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 234–247. ACM, 2012.

10 Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 2011.

11 Jimmy Chan, Fei Li, and Yun Wang. Discriminatory persuasion: How to convince a group. *Available at SSRN*, 2015.

12 Vincent P Crawford and Joel Sobel. Strategic information transmission. *Econometrica: Journal of the Econometric Society*, 1982.

13 Shaddin Dughmi. Submodular functions: Extensions, distributions, and algorithms. a survey. *arXiv preprint arXiv:0912.0322*, 2009.

14 Shaddin Dughmi, Nicole Immorlica, and Aaron Roth. Constrained signaling in auction design. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1341–1357. Society for Industrial and Applied Mathematics, 2014.

15 Shaddin Dughmi and Haifeng Xu. Algorithmic bayesian persuasion. *STOC 2016*, 2016.

16 Yuval Emek, Michal Feldman, Iftah Gamzu, Renato PaesLeme, and Moshe Tennenholtz. Signaling schemes for revenue maximization. *ACM Transactions on Economics and Computation*, 2(2):5, 2014.

17 Uriel Feige, Magnús M Halldórsson, Guy Kortsarz, and Aravind Srinivasan. Approximating the domatic number. *SIAM Journal on computing*, 2002.

18 Matthew Gentzkow and Emir Kamenica. Bayesian persuasion. *American Economic Review*, 2011.

19 Matthew Gentzkow and Emir Kamenica. Competition in persuasion. Technical report, National Bureau of Economic Research, 2011.

20 Matthew Gentzkow and Emir Kamenica. Costly persuasion. *The American Economic Review*, 2014.

21 Itay Goldstein and Yaron Leitner. Stress tests and information disclosure. Technical report, FRB of Philadelphia Working Paper, 2015.

22 Mingyu Guo and Argyrios Deligkas. Revenue maximization via hiding item attributes. *arXiv preprint arXiv:1302.5332*, 2013.

23 David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

24 Subhash Khot, Richard J Lipton, Evangelos Markakis, and Aranyak Mehta. Inapproximability results for combinatorial auctions with submodular utility functions. In *Internet and Network Economics*. Springer, 2005.

25 Kazuo Murota. Convexity and steinitz's exchange property. In *Integer Programming and Combinatorial Optimization*, pages 260–274. Springer, 1996.

26 Kazuo Murota. *Discrete convex analysis*. SIAM, 2003.

27 Keith E Schnakenberg. Expert advice to a voting body. *Journal of Economic Theory*, 2015.

28 Wataru Tamura. Optimal monetary policy and transparency under informational frictions. *Available at SSRN 2191900*, 2014.

29 Ina A Taneva. Information design. Technical report, University of Edinburgh, 2015.

30 Jan Vondrák. *Submodularity in combinatorial optimization*. PhD thesis, PhD thesis, Charles University, Prague, Czech Republic, 2007.

31 Yun Wang. Bayesian persuasion with multiple receivers. *Available at SSRN 2625399*, 2013.

Condorcet-Consistent and Approximately Strategyproof Tournament Rules

Jon Schneider[1], Ariel Schvartzman[2], and S. Matthew Weinberg[3]

1 **Department of Computer Science, Princeton University, Princeton, USA**
`js44@cs.princeton.edu`
2 **Department of Computer Science, Princeton University, Princeton, USA**
`acohenca@cs.princeton.edu`
3 **Department of Computer Science, Princeton University, Princeton, USA**
`smweinberg@princeton.edu`

Abstract

We consider the manipulability of tournament rules for round-robin tournaments of n competitors. Specifically, n competitors are competing for a prize, and a tournament rule r maps the result of all $\binom{n}{2}$ pairwise matches (called a *tournament*, T) to a distribution over winners. Rule r is *Condorcet-consistent* if whenever i wins all $n-1$ of her matches, r selects i with probability 1.

We consider strategic manipulation of tournaments where player j might throw their match to player i in order to increase the likelihood that one of them wins the tournament. Regardless of the reason why j chooses to do this, the potential for manipulation exists as long as $\Pr[r(T) = i]$ increases by more than $\Pr[r(T) = j]$ decreases. Unfortunately, it is known that every Condorcet-consistent rule is manipulable [1]. In this work, we address the question of *how manipulable* Condorcet-consistent rules must necessarily be - by trying to minimize the difference between the increase in $\Pr[r(T) = i]$ and decrease in $\Pr[r(T) = j]$ for any potential manipulating pair.

We show that every Condorcet-consistent rule is in fact 1/3-manipulable, and that selecting a winner according to a random single elimination bracket is not α-manipulable for any $\alpha > 1/3$. We also show that many previously studied tournament formats are all 1/2-manipulable, and the popular class of Copeland rules (any rule that selects a player with the most wins) are all in fact 1-manipulable, the worst possible. Finally, we consider extensions to match-fixing among sets of more than two players.

1998 ACM Subject Classification J.4 Social and Behavioral Sciences

Keywords and phrases Tournament design, Non-manipulability, Condorcet-consistent, Strategyproofness

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.35

1 Introduction

In recent years, numerous scandals have unfolded surrounding match fixing and throwing at the highest levels of competitive sports (e.g. Olympic Badminton [11], Professional Tennis [6], European Football [22], and even eSports [24]). In some instances, the motivation behind these scandals was gambling profits, and no amount of clever tournament design can possibly mitigate this. In others, however, the surprising motivation was an improved performance *at that same tournament.* For instance, four Badminton teams (eight players) were disqualified from the London 2012 Olympics for throwing matches. Interestingly, the reason teams wanted to lose their matches was in order to *improve* their probability of winning an Olympic medal. Olympic Badminton (like many other sports) conducts a two-phase tournament. In the first stage, groups of four play a round-robin tournament, with the top two teams advancing. In


© Jonathan Schneider, Ariel Schvartzman, and Seth Matthew Weinberg;
licensed under Creative Commons License CC-BY

the second stage, the advancing teams participate in a single elimination tournament, seeded according to their performance in the group stage. An upset in one group left one of the world's top teams with a low seed, so many teams actually preferred to receive a *lower* seed coming out of the group stage to face the tougher opponent as late as possible.

While much of the world blames the teams for their poor sportsmanship, researchers in voting theory have instead critiqued the poor tournament design that punished teams for trying to maximize their chances of winning a medal. Specifically, the two-phase tournament lacks the basic property of *monotonicity*, where no competitor can unilaterally improve their chances of winning by throwing a match that they otherwise could have won. Thus, recent work has addressed the question of whether tournament structures exist that are both fair, in that they select some notion of a qualified winner, and strategyproof, in that teams have no incentive to do anything but play their best in each match.

One minimal notion of fairness studied is *Condorcet-consistence*, which just guarantees that whenever one competitor wins *all* of their matches (and is what's called a *Condorcet winner*), they win the event with probability 1. Designing Condorcet-consistent, monotone rules is simple: any single elimination bracket suffices. Popular voting rules such as the Copeland Rule or the Random Condorcet Removal Rule are also Condorcet-consistent and monotone, but two-phase tournaments with an initial group play aren't [17].

Still, monotonicity only guarantees that no team wishes to unilaterally throw a match to improve their chances of winning, whereas one might also hope to guarantee that no two teams could fix the outcome of their match in order to improve the probability that one of them wins. While we have to go back further in history to find a clear instance of this kind of match-fixing, it did indeed result in a historical scandal. In the 1982 FIFA World Cup (again a two-stage tournament), Austria, West Germany, and Algeria were in the same group of four where two would advance. Algeria had already won two matches and lost one, Austria was 2-0, West Germany was 1-1, and the only remaining game was Austria vs. West Germany. Due to tie-breakers and the specific outcomes of previous matches, Austria would have been eliminated by a large West German victory, and West Germany would have been eliminated by a loss or draw. Once West Germany scored an early goal, *both* teams essentially threw the rest of the match, allowing both of them to advance at Algeria's expense [25]. While the incident was never formally investigated, many fans were confident the teams had colluded beforehand, and the event is remembered as the "disgrace of Gijón." Before being eliminated, Algeria had become the first African team to beat a European team at the World Cup, and also the first to win two games. West Germany went on to become the runners-up of the tournament.

Motivated by events like this, it is important also to design tournaments where no two teams can fix the outcome of their match and improve the probability that one of them wins. Altman and Kleinberg terms this property 2-Strongly Nonmanipulable (2-SNM), and showed that no tournament rule is both Condorcet-consistent and 2-SNM [1] (it was previously shown by Altman et. al. that no *deterministic* rule is both Condorcet-consistent and 2-SNM [2]).

In light of this, both works relax the notion of Condorcet-consistency and design tournament rules that are at least *non-imposing* (could possibly select each competitor as a winner) and 2-SNM [2], or α-Condorcet-consistent (if there is a Condorcet winner, she wins with probability at least α) and 2-SNM. While these relaxations are well-motivated for settings where pair-wise comparisons are only *implicitly* made, and not even necessarily learned in the end (e.g. elections), it is hard to imagine a successful sports competition format where a competitor could win all their matches and still leave empty handed. This happened during the 2008 NCAA Football Season. Utah went undefeated (#2, 13-0) in their region but

were not invited to the bowl game because critics deemed their schedule weak. They were eventually ranked second nation-wide and beat Alabama (#6, 12-2) in the Sugar Bowl, while Florida (#1, 13-1) beat Oklahoma (#5, 12-2) for the National Championship. This event prompted organizers to reconsider the process by which teams are invited to the National Championship game.

Motivated by match-based applications such as sporting events, where the outcome of pair-wise matches is *explicitly* learned and used to select a winner, we consider instead the design of tournament rules that are exactly Condorcet-consistent, but only approximately 2-SNM. Specifically, we say that a tournament rule is 2-SNM-α if it is *never* possible for two teams i and j to fix their match such that the probability that the winner is in $\{i, j\}$ improves by at least α. The idea behind this relaxation is that whatever motivates j to throw the match (perhaps j and i are teammates, perhaps i is paying j some monetary bribe, etc.), the potential gains scale with α. So it is easier to disincentivize manipulation (either through investigations and punishments, reputation, or just feeling morally lousy) in tournaments that are less manipulable.

1.1 Our Results

Our main result is a matching upper and lower bound of 1/3 on attainable values of α for Condorcet-consistent 2-SNM-α tournament rules. The optimal rule that attains this upper bound is actually quite simple: a random single elimination bracket. Specifically, each competitor is randomly placed into one of $2^{\lceil \log_2 n \rceil}$ seeds, along with $2^{\lceil \log_2 n \rceil} - n$ byes, and then a single elimination tournament is played.

Proving a lower bound of 1/3 is straight-forward: imagine a tournament with three players, A, B and C, where A beats B, B beats C, and C beats A. Then some pair must win with combined probability at most 2/3. Yet, any pair could create a Condorcet winner by colluding, who necessarily wins with probability 1 in any Condorcet-consistent rule. Embedding this within examples for arbitrary n is also easy: just have A, B, and C each beat all of the remaining $n - 3$ competitors[1].

On the other hand, proving that a random single elimination bracket is optimal is tricky, but our proof is still rather clean. For any i, j in any tournament, we directly show that i can improve her probability of winning by at most 1/3 when j throws their match using a coupling argument. For every deterministic single elimination bracket where i and j could potentially gain from manipulation (because i would be the champion if i beat j, but j would *not* be the champion even if j beat i), we construct *two* deterministic single elimination brackets where no potential exists (possibly because one of them will lose before facing each other, or because the winner would be in $\{i, j\}$ no matter the outcome of their match). For our coupling to be valid, we not only need each mapping to be invertible, but also for their images to be disjoint. Our coupling is necessarily somewhat involved in order to obtain this property, but otherwise we believe our proof is likely as simple as possible. Because the probability that j wins cannot possibly go up by throwing a match to i, this immediately proves that a random single elimination bracket is 2-SNM-1/3.

We also show that the Copeland rule, a popular rule that chooses the team with the most wins, is asymptotically 2-SNM-1, the *worst* possible. Essentially, the problem is that if all teams have the same number of wins, then any two can collude to guarantee that one of

[1] Interestingly, this lower-bound example is far from pathological and occurs at even the highest levels of professional sports (see [18], for instance).

them wins, no matter the tie-breaking rule. We further show that numerous other formats, (the Random Voting Caterpillar, the Iterative Condorcet Rule, and the Top Cycle Rule) are all at best 2-SNM-1/2. The same example is bad for all three formats: there is one superman who beats $n-2$ of the remaining players, and one kryptonite, who beats only the superman (but loses to the other $n-2$ players).

Our results extend to settings where the winner of each pairwise match is not deterministically known, but randomized (i.e. all partipants know that i will beat j with probability p_{ij}). Specifically, we show that any rule that is 2-SNM-α when all $p_{ij} \in \{0,1\}$ is also 2-SNM-α for arbitrary p_{ij}. Clearly, any lower bound using integral p_{ij} also provides a lower bound for arbitrary p_{ij}, so as far as upper/lower bounds are concerned the models are equivalent. Of course, the randomized model is much more realistic, so it is convenient that we can prove theorems in this setting by only studying the deterministic setting, which is mathematically much simpler.

Finally, we consider manipulations among coalitions of $k > 2$ participants. We say that a rule is k-SNM-α if no set S of size $\leq k$ can *ever* manipulate the outcomes of matches between players in S to improve the probability that the winner is in S by more than α. We prove a simple lower bound of $\alpha = \frac{k-1}{2k-1}$ on all Condorcet-consistent rules, and conjecture that this is tight.

1.2 Related Works

The mathematical study of tournament design has a rich literature, ranging from social choice theory to psychology. The overarching goal in these works is to design tournament rules that satisfy various properties a designer might find desirable. Examples of such properties might be that all players are treated equally, that a winner is chosen without a tiebreaking procedure, or that a "most qualified" winner is selected [8, 20, 7, 19, 28, 15, 23]. See [14] for a good review of this literature and its connections to other fields as well.

Most related to our work are properties involving *strategic manipulation*. In the more general field of Voting Theory, there is a rich literature on the design of strategyproof mechanisms dating back to Arrow's Impossibility Theorem [3] and the Gibbard-Satterthwaite Theorem [9, 21, 10]. While tournaments are a very special case (voters are indifferent among outcomes where they do not win, voters can only "lie" in specific ways, etc.), tournament design indeed seems to inherit much of the impossibility associated with strategyproof voting procedures [1], [2].

Specifically, Altman et. al. proved that no deterministic tournament rule is 2-SNM and Condorcet-consistent, and Altman and Kleinberg proved that no randomized tournament rule is 2-SNM and Condorcet-consistent either [2, 1]. More recently, Pauly studied the specific two-stage tournament rule used by the World Cup (and Olympic Badminton, etc.) [17]. There, it is shown essentially that the problem lies in the first round group stage: no changes to the second phase can possibly result in a strategyproof [2] tournament.

To cope with their impossibility results, Altman et. al. propose a relaxation of Condorcet-consistence called *non-imposing*. A rule r is non-imposing if for all i, there exists a T such that player i wins with probability 1. They design a clever recursive rule that is non-imposing and 2-SNM for all $n \neq 3$. Interestingly, they also show that for $n = 3$ no such rule exists. Altman and Kleinberg consider a different relaxation called α-*Condorcet-consistent*. A rule r is α-Condorcet-consistent if whenever i is a Condorcet winner in T, we have their probability of winning T is at least α. They design a rule that is $2/n$-Condorcet-consistent and 2-SNM (in fact it is also k-SNM for all k), but conjecture that much better is attainable.

[2] See [17] for the specific notion of strategyproofness studied.

The two works above are most similar to ours in spirit: motivated by the non-existence of Condorcet-consistent and 2-SNM tournament rules, we relax one of the notions. These previous works relax Condorcet-consistency while maintaining 2-SNM exactly, and are most appropriate in settings where pairwise comparisons of players are only learned *implicitly* (or perhaps not at all) through the outcome and not *explicitly* as the result of matches. Instead, we relax the notion of 2-SNM and maintain the notion of Condorcet-consistency exactly. In settings like sports competitions where pairwise comparisons of players are learned explicitly through matches played, Condorcet-consistency is a non-negotiable desideratum. Therefore, we believe our approach is more natural in such settings.

Another line of work introduced by [4] considers a different kind of strategyproofness: how much control does the designer of a single-elimination tournament have over the winner? Can the designer efficiently find a bracket in such a way to maximize the likelihood that a player of their choice wins the tournament? The models in this area assume that the designer is given the probabilities p_{ij} that team i beats team j and the problem is known in the literature as *agenda control* when p_{ij} are real numbers and Tournament Fixing Problem (TFP) when all probabilities are 0 or 1.

On the negative side, it is known that for n-player tournaments it is NP-hard to decide whether or not there exists a seeding such that the probability of team k winning is at least δ, given k, δ, even if $\mathsf{p}_{ij} \in \{0, 0.5, 1\}$ for all i, j [27]. [26] show that the hardness results persist even for the TFP when the given team k is a king (for every team j, either k beats j or k beats a team that beats j) with at least $n/4$ wins, or a 3−king (is at most 3 "wins" away from every team) that wins at least half of their games. Follow up work [13] shows that in the case of balanced single elimination brackets, it is still NP-hard to find a bracket that favors team k when the designer is allowed to bribe at most $(1 - \varepsilon) \log n$ of the teams to throw their respective matches.

On the positive side, there exist structural results that dictate when it is computationally efficient to find a tournament that favors a given team. [26] show conditions under which, for large enough tournaments, any sufficiently good team can be favored by the tournament seeding. Other results [13, 12] show conditions under which 3−kings can be made into winners of single-elimination tournaments.

A large body of literature exists regarding manipulation and bribery in the context of voting rules. For an introduction, we recommend the reader consult chapter 7 of the handbook [16].

1.3 Conclusions and Future Work

Our work contributes to a recent literature on incentive compatible tournament design. While most previous works insist on strong incentive properties and relaxed fairness properties, such rules are inadequate for sporting events. Instead, we insist at least that events maintain Condorcet-consistency, and aim to relax strategyproofness as minimally as possible.

At a high level, our work suggests (similar to previous works), that single elimination brackets are desirable whenever incentive issues come into play. However, previous desiderata (such as those considered in [1]) don't necessarily rule out other tournament formats, like the Copeland rule, which is ubiquitous in tournaments (both as a complete format and as subtournaments in a two-phase format). In comparison, our work identifies single elimination brackets (2-SNM-1/3) as having significantly better strategic properties versus the Copeland rule (2-SNM-1).

Our work also identifies two practical suggestions when match-fixing is a concern that aren't explained by prior benchmarks. First, when hosting a single elimination tournament,

it might be desirable to release the exact bracket as late as possible. The idea is that as soon as the exact bracket is known, competitors have greater incentive to fix matches (in our model, up to three times as much), which presumably takes some time and organization. Obviously, there are more tradeoffs at play: a later release inconveniences athletes and fans, and (perhaps more importantly to the designers) could negatively impact ticket sales. But our work does at least identify match-fixing as a part of this tradeoff. Note that some Olympic events (such as Taekwondo) contest the entire competition in a single day at a single venue, so a delayed release may indeed be practical. We also note that a similar "fix" was applied after the 1982 World Cup: the last two matches in each group are now played at the same time to minimize the amount of information teams have when making potentially strategic decisions.

Additionally, our work suggests that even in the optimal tournament, hefty punishments for cheaters might be necessary in order to discourage match-fixing (even without taking gambling into consideration). In many sports, winning an Olympic gold can make a career. Unfortunately, our work suggests that punishments roughly on this order might be necessary in order to properly deter match-fixing.

Finally, we propose two directions for future work. First, while we obtain tight results for Condorcet-consistent 2-SNM-α rules, we only prove a lower bound of k-SNM-$\frac{k-1}{2k-1}$ for Condorcet-consistent rules and $k > 2$. We conjecture that this is tight, but unfortunately simulations indicate that all of the formats studied in our work do *not* achieve this bound. So it is an interesting open question to design a rule that does. Even partial results (of the form identified below) would require a new tournament format than those considered in this work.

▶ **Open Question 1.** *Does there exist a tournament rule that is Condorcet-consistent and k-SNM-$\frac{k-1}{2k-1}$ for all k? What about a family of rules $\mathcal{F}$ such that for all k, F_k is k-SNM-$\frac{k-1}{2k-1}$? What about a rule that is k-SNM-$1/2$ for all k?*[3]

It is also important to study what bounds are attainable in restricted versions of our probabilistic model (e.g. if for all i, j, the probability that i beats j lies in $[\epsilon, 1-\epsilon]$). Realistic instances at least have *some* non-zero probability of an upset in every match, but our lower bounds don't hold in this model. So it is interesting to see if better formats are possible.

▶ **Open Question 2.** *Is a random single elimination bracket still optimal among Condorcet-consistent rules (w.r.t. 2-SNM-α) if for all i, j, the probability that i beats j lies in $[\epsilon, 1-\epsilon]$? How does the optimal attainable α for Condorcet-consistent, 2-SNM-α tournament formats change as a function of ϵ?*

2 Preliminaries and Notation

In this section, we present notation used throughout the remainder of the paper. Where possible, we adopt notation from [1].

▶ **Definition 1.** A (round-robin) *tournament* T on n players is the set of outcomes of the $\binom{n}{2}$ games played between all pairs of distinct players. We write $T_{ij} = 1$ if player i beats player j and $T_{ij} = -1$ otherwise. We also let $\mathcal{T}_n$ denote the set of tournaments on n players.

▶ **Definition 2.** For a subset $S \subseteq [n]$ of players, two tournaments T and T' are *S-adjacent* if they only differ on the outcomes of some subset of games played between members of S.

[3] Note that $\frac{k-1}{2k-1} \to 1/2$ as $k \to \infty$.

In particular, two tournaments T and T' are $\{i, j\}$ adjacent if they only differ in the result of the game played between player i and player j.

▶ **Definition 3.** A *tournament rule* (or *winner determination rule*) $r : \mathcal{T}_n \to \Delta([n])$ is a mapping from the set of tournaments on n players to probability distributions over these n players (representing the probability we choose a given player to be the winner). We will write $r_i(T) = \Pr[r(T) = i]$ to denote the probability that player i wins tournament T under rule r.

Many tournament rules, while valid by the above definition, would be ill-suited for running an actual tournament; for example, the tournament rule which always crowns player 1 the winner. In an attempt to restrict ourselves to 'reasonable' tournament rules, we consider tournaments that obey the following two criteria.

▶ **Definition 4.** Player i is a *Condorcet winner* in tournament T if player i wins their match against all the other $n - 1$ players. A tournament rule r is *Condorcet-consistent* if $r_i(T) = 1$ whenever i is a Condorcet winner in T.

▶ **Definition 5.** A tournament rule r is *monotone* if, for all i, $r_i(T)$ does not increase when i loses a game it wins in T. That is, if i beats j in T and T and T' are $\{i, j\}$ adjacent, then if r is monotone, $r_i(T) \geq r_i(T')$.

Intuitively, this first criterion requires us to award the prize to the winner in the case of a clear winner (hence making the tournament a contest of skill), and the second criterion makes it so that players have an incentive to win their games. There are various other criteria one might wish a tournament rule to satisfy; many can be found in [1].

In this paper, we consider the scenario where certain coalitions of players attempt to increase the overall chance of one of them winning by manipulating the outcomes of matches within players of the coalition. The simplest case of this is in the case of coalitions of size 2, where player j might throw their match to player i. If T is the original tournament and T' is the manipulated tournament where j loses to i, then player i gains $r_i(T') - r_i(T)$ from the manipulation, and player j loses $r_j(T) - r_j(T')$ (in terms of probability of winning). Therefore, as long as $r_i(T') - r_i(T) > r_j(T) - r_j(T')$, it will be in the players' interest to manipulate. Equivalently, if $r_i(T') + r_j(T') > r_i(T) + r_j(T)$ (i.e., the probability either player i or j wins increases upon throwing the match), there is incentive for i and j to manipulate.

Ideally, we would like to choose a tournament rule so that, regardless of the tournament, there will be no incentive to perform manipulations of the above sort. This is encapsulated in the following definition from [1].

▶ **Definition 6.** A tournament rule r is *2-strongly non-manipulable (2-SNM)* if, for all pairs of $\{i, j\}$-adjacent tournaments T and T', $r_i(T) + r_j(T) = r_i(T') + r_j(T')$.

Unfortunately, no tournament rules exist that are simultaneously Condorcet-consistent and 2-strongly non-manipulable (this is shown in [1] and also follows from our lower bound in Section 3.1). As tournament designers, one way around this obstacle is to discourage manipulation. This discouragement can take many forms, both explicit (if players are caught fixing matches, they are disqualified/fined) and implicit (it is logistically hard to fix matches, it is unsportsmanlike). So the focus of this paper is to quantify *how manipulable* certain tournament formats are (i.e. how much can teams possibly gain by fixing matches), the idea being that it is easier to discourage manipulation in tournaments that are less manipulable.

▶ **Definition 7.** A tournament rule r is *2-strongly non-manipulable at probability α (2-SNM-α)* if, for all i and j and pairs of $\{i, j\}$-adjacent tournaments T and T', $r_i(T') + r_j(T') - r_i(T) - r_j(T) \leq \alpha$.

It is straightforward to generalize this definition to larger coalitions of colluding players.

▶ **Definition 8.** A tournament rule r is *k-strongly non-manipulable at probability α (k-SNM-α)* if, for all subsets S of players of size at most k, for all pairs of S-adjacent tournaments T and T', $\sum_{i \in S} r_i(T') - \sum_{i \in S} r_i(T) \leq \alpha$.

2.1 The Random Single-Elimination Bracket Rule

Our main result concerns a specific tournament rule we call the *random single-elimination bracket rule*. This rule can be defined formally as follows.

▶ **Definition 9.** A *single-elimination bracket* (or *bracket*, for short) B on $n = 2^h$ players is a complete binary tree of height h whose leaves are labelled with some permutation of the n players. The outcome of a bracket B under a tournament T is the labelling of internal nodes of B where each node is labelled by the winner of its two children under T. The winner of B under T is the label of the root of B under this labelling.

▶ **Definition 10.** The *random single-elimination bracket rule* r is a tournament rule on $n = 2^h$ players where $r_i(T)$ is the probability player i is the winner of B under T when B is chosen uniformly at random from the set of $n!$ possible brackets.

If n is not a power of 2, we define the random single-elimination bracket rule on n players by introducing $2^{\lceil \log_2 n \rceil} - n$ dummy players who lose to all of the existing n players.

It is straightforward to check that the random single-elimination bracket rule is both Condorcet-consistent and monotone. Our main result (Theorem 13) shows that in addition to these properties, the random single-elimination bracket rule is 2-SNM-1/3 (which is the best possible, by Theorem 11).

We give some examples of other common tournament rules in Section 3.4. While many of these rules are both Condorcet-consistent and monotone, we do not know of any which are additionally 2-SNM-1/3.

3 Main Result

3.1 Lower bounds for k-SNM-α

We begin by showing that no tournament rule is 2-SNM-α for $\alpha < 1/3$. A similar theorem appears as Proposition 17 in [1] (which states that $\alpha = 0$ is impossible).

▶ **Theorem 11.** *There is no Condorcet-consistent tournament rule on n players (for $n \geq 3$) that is 2-SNM-α for $\alpha < \frac{1}{3}$.*

Proof. Consider the tournament T on three players A, B, and C where A beats B, B beats C, and C beats A (illustrated in Figure 1). . Note that, while this tournament has no Condorcet winner, changing the result of any of the three games results in a Condorcet winner. For example, if A bribes C to lose to A, then A becomes the Condorcet winner.

If we have a tournament rule r that is 2-SNM-α, then combining this with the above fact gives rise to the following three inequalities.

$$\begin{aligned} r_A(T) + r_B(T) &\geq 1 - \alpha \\ r_B(T) + r_C(T) &\geq 1 - \alpha \\ r_C(T) + r_A(T) &\geq 1 - \alpha \end{aligned}$$

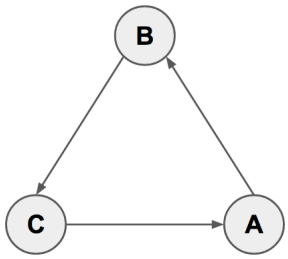


Figure 1 A tournament which attains the lower bound of $\alpha = 1/3$ for all tournament rules.

Together these imply $r_A(T)+r_B(T)+r_C(T) \geq \frac{3}{2}(1-\alpha)$. But $r_A(T)+r_B(T)+r_C(T) = 1$; it follows that $\alpha \geq \frac{1}{3}$, as desired.

We can extend this counterexample to $n > 3$ players by introducing $n-3$ dummy players who all lose to A, B, and C; the argument above continues to hold. ◀

We can use similar logic to prove lower bounds for the more general case of k-SNM-α.

▶ **Theorem 12.** *There is no Condorcet-consistent tournament rule on n players (for $n \geq 2k-1$) that is k-SNM-α for $\alpha < \frac{k-1}{2k-1}$.*

Proof. Consider the following tournament T on the $2k-1$ players labelled 1 through $2k-1$. Each player i wins their match versus the $k-1$ players $i+1, i+2, \ldots, i+(k-1)$, and loses their match versus the $k-1$ players $i-1, i-2, \ldots, i-(k-1)$ (indices taken modulo $2k-1$). Note that the coalition of players $S_i = \{i, i-1, \ldots, i-(k-1)\}$ of size k can cause i to become a Condorcet winner if all players in the coalition agree to lose their games with i. If we have a tournament rule r that is k-SNM-α, then this implies the following $2k-1$ inequalities (one for each $i \in [2k-1]$):

$$\sum_{j \in S_i} r_j(T) \geq 1 - \alpha \tag{1}$$

Summing these $2k-1$ inequalities, we obtain

$$k \sum_{j=1}^{2k-1} r_j(T) \geq (2k-1)(1-\alpha) \tag{2}$$

Since $\sum_{j=1}^{2k-1} r_j(T) \leq 1$, this implies that $\alpha \geq \frac{k-1}{2k-1}$, as desired. Again, it is possible to extend this example to any number of players $n \geq 2k-1$ by introducing dummy players who lose to all $2k-1$ of the above players. ◀

3.2 Random single elimination brackets are 2-SNM-1/3

We now show that the random single elimination bracket rule is optimal against coalitions of size 2. The proof idea is simple; for every bracket B that contributes to the incentive to manipulate $r_i(T') + r_j(T') - r_i(T) - r_j(T)$ we will show that there are two that do not (in other words, for every scenario where team i benefits from the manipulation, there exist two other scenarios where the maniuplation does not benefit either team).

▶ **Theorem 13.** *The random single elimination bracket rule is 2-SNM-1/3.*

Proof. Let $\mathcal{B}$ be the set of $n!$ different possible brackets amongst the n players. For a given tournament T and a given player i, write $\mathbb{1}(B, T, i)$ to represent the indicator variable which is 1 if i wins bracket B under the outcomes in T and 0 otherwise. Then we can write

$$r_i(T) = \frac{1}{|\mathcal{B}|} \sum_{B \in \mathcal{B}} \mathbb{1}(B, T, i).$$

Assume i loses to j in T. Then, if we let T' be the tournament that is $\{i, j\}$ adjacent to T, we can write the increase in utility resulting from j throwing to i

$$\frac{1}{|\mathcal{B}|} \sum_{B \in \mathcal{B}} \left(\mathbb{1}(B, T', i) + \mathbb{1}(B, T', j) - \mathbb{1}(B, T, i) - \mathbb{1}(B, T, j) \right). \tag{3}$$

Our goal is to show that this sum is at most 1/3. Now, note that if i does not end up playing j in bracket B under T, i also does not play j in B under T' (and vice versa). In these brackets, $\mathbb{1}(B, T', i) = \mathbb{1}(B, T, i)$ and $\mathbb{1}(B, T', j) = \mathbb{1}(B, T, j)$, so these brackets contribute nothing to the sum in Equation 3. On the other hand, in a bracket B where i does play j, we are guaranteed that $\mathbb{1}(B, T, i) = 0$ and $\mathbb{1}(B, T', j) = 0$ (since i loses to j in T and j loses to i in T'). Therefore, letting $\mathcal{B}_{ij}$ be the subset of $\mathcal{B}$ of brackets where i meets j, we can rewrite Equation 3 as

$$\frac{1}{|\mathcal{B}|} \sum_{B \in \mathcal{B}_{ij}} \left(\mathbb{1}(B, T', i) - \mathbb{1}(B, T, j) \right).$$

Since $\mathbb{1}(B, T', i) \leq 1$, this is at most

$$\frac{1}{|\mathcal{B}|} \sum_{B \in \mathcal{B}_{ij}} \left(1 - \mathbb{1}(B, T, j) \right).$$

This final sum counts exactly the number of brackets B where i and j meet (under T, so j beats i) but j does not win the tournament. Call such brackets *bad*, and call the remaining brackets *good*. We will exhibit two injective mappings σ_i and σ_j from bad brackets to good brackets such that the ranges of σ_i and σ_j are disjoint. This implies that there are at least twice as many good brackets as bad brackets, and thus that the sum above is at most 1/3, completing the proof.

For both mappings, we will need the following terminology. Consider a bad bracket B, and consider the path from j up to the root of this tree. The nodes of this path are labelled by players that j would face if they got that far. More specifically, j has some opponent in the first round. Should j win, j would face some opponent in the second round, then the third round, etc. all the way to the finals, and these opponents do not depend on the outcomes of any of j's matches. Then since B is a bad bracket, j does not win, and at least one of the players on this path can beat j. Choose the **latest** such player (i.e. the closest to the root) and call this player k. Note that k might *not* be the player that knocks j out of the tournament (that is the *first* player along this path who would beat j).

Suppose that i and j meet at height h of the bracket (i.e. in the h^{th} round). Let B_i, B_j, B_k be the subtrees of height h that contain i, j, and k respectively. An example is shown in Figure 2.

We first describe the simpler of the two maps, σ_i. Define $\sigma_i(B)$ by swapping the subtrees B_i and B_k as shown in Figure 3. In this bracket j will lose to k before ever meeting i, so $\sigma_i(B)$ is good. Moreover σ_i is injective since we can construct its inverse. In $\sigma_i(B)$, j certainly would lose to k at height h before reaching i. Furthermore, because we didn't

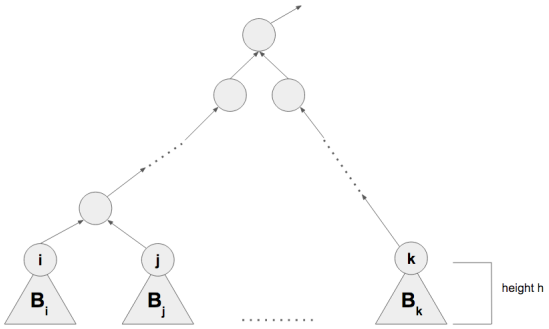


Figure 2 An example of a bad bracket B.

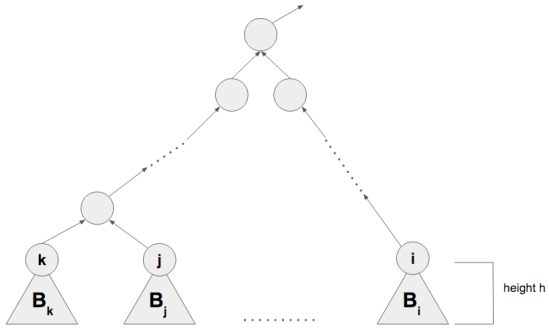


Figure 3 $\sigma_i(B)$.

change B_j at all, j still wins all of its first $h-1$ matches and makes it to k (because we started from a B where j makes it to i at height h). So we can identify k as the first player who beats j in $\sigma_i(B)$, learn the height h, and undo the swap of B_k and B_i.

We now describe the second map, σ_j. To construct $\sigma_j(B)$, begin by swapping the subtrees B_j and B_k (see Figure 4). Note that the bracket formed in this way is good; since we chose k to be the latest player on j's path to victory that can beat j, if j meets i, j will also beat all subsequent players and win the tournament (note that it is of course possible that j doesn't even make it to i, in which case $\sigma_j(B)$ is still good. But it is clear that *if* j meets i, then j will win the tournament, so $\sigma_j(B)$ is good in either case). Unfortunately, this map as stated is not injective; in particular, we cannot recover the height h to undo the swap as in the previous case.

The only reason we cannot uniquely identify k in the same way as when we invert σ_i is that i might meet some player k' at height $h' < h$ in B_i who also could beat j. So, intuitively, we would like to swap such players out with players who lose to j. Since j beats all of its opponents in B_j, B_j is an ample source of such players. We will therefore perform some additional 'subswap' operations, swapping subtrees of B_j and B_k so as to uniquely identify k as the first player i meets in $\sigma_j(B)$ who can beat j.

Specifically, for $0 \le h' < h$, let $a(h')$ be the opponent i plays at height h' in B_i, and let $B_i(h')$ be the subtree of B_i with root $a(h')$ (note that the player that i meets at height h' is the root of a subtree of height $h'-1$, and that all these subtrees are disjoint). Similarly, let $b(h')$ be the opponent j plays at height h' in B_j, and let $B_j(h')$ be the subtree of B_j with root $b(h')$. To construct $\sigma_j(B)$ from B, first swap B_j and B_k. Then for each $h' \in [0, h)$ such that $a(h')$ would beat j, swap the subtrees $B_i(h')$ and $B_j(h')$. See Figure 5 for an illustration of a subswap operation.

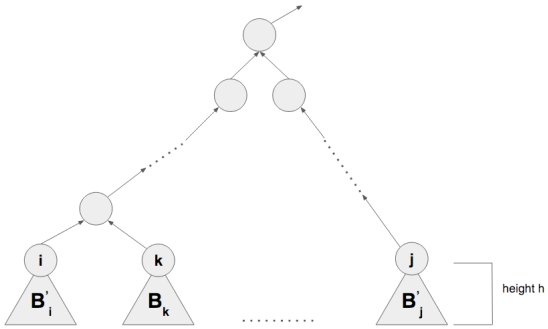


Figure 4 $\sigma_j(B)$.

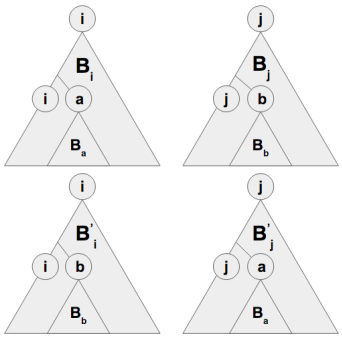


Figure 5 Subswap operation for σ_j.

Note that $\sigma_j(B)$ is still good; it is still the case that if j meets i, j will beat all subsequent players (all we have done in that part of the bracket is perhaps alter whether or not j will indeed meet i). On the other hand, since j makes it to height h in B_j, j can beat player $b(h')$ for all h', so k is now the first player i would encounter in $\sigma_j(B)$ who can beat j. From this, we can recover k and thus h, and undo the swap of B_i and B_j. To undo the subswaps, observe that because we started with a bad bracket B, that j must have beaten all opponents it faces in the first h rounds. Since all opponents on j's path who beat j at height less than h were necessarily put there by our subswap operations, we can just find all such opponents and swap them back out. This process inverts σ_j, thus proving that σ_j is injective.

Finally, note that in $\sigma_i(B)$, k must play j before either plays i, whereas in $\sigma_j(B)$, k must play i before either plays j. Therefore the ranges of σ_i and σ_j are disjoint, and this completes the proof.

For the reader aiming to understand our coupling argument better, Appendix A contains some specific examples. ◀

3.3 Extension to randomized outcomes

Thus far we have been assuming that all match results are deterministic and known to the players in advance. Of course, this is not true in general; in real life, the outcomes of games are inherently unpredictable. It is perhaps imaginable that this unpredictability could increase the incentive to manipulate. In this section we show that this is not the case; a simple application of linearity of expectation shows that results about deterministic tournaments still hold for their randomized counterparts. We begin by defining a randomized tournament as follows.

▶ **Definition 14.** A *randomized tournament* $\mathcal{T}$ is a random variable whose values range over (deterministic) tournaments T. As shorthand, we will write $\mathbb{P}_{\mathcal{T}}(T)$ to represent the probability that $\mathcal{T} = T$.

Note that this definition accounts for the most straightforward generalization of tournament outcomes from deterministic to randomized, where for each match between players i and j we assign a probability p_{ij} to the probability that i beats j. This definition further allows for the possibility of correlation between matches (e.g., with some probability player i has a good day and wins all his matches, and with some probability he has a bad day and loses all his matches).

Manipulations in this randomized model are similar to manipulations in the deterministic model in that they effectively force the result of a match to a win or a loss. Formally, let $\sigma_{ij}(T)$ for a (deterministic) tournament T be the tournament formed by T but where i beats j (if i beats j in T, then $\sigma_{ij}(T) = T$). A tournament rule r is 2-SNM-α if for all i and j,

$$\mathbb{E}_{\mathcal{T}}\left[r_i(\sigma_{ij}(\mathcal{T})) + r_j(\sigma_{ij}(\mathcal{T})) - r_i(\mathcal{T}) - r_j(\mathcal{T})\right] \leq \alpha \tag{4}$$

We then have the following theorem:

▶ **Theorem 15.** *If a rule r is 2-SNM-α in the deterministic tournament model, it is also 2-SNM-α in the randomized tournament model.*

Proof. Note that we can write the expectation in Equation 4 as

$$\sum_{T} \mathbb{P}_{\mathcal{T}}(T)\left(r_i(\sigma_{ij}(T)) + r_j(\sigma_{ij}(T)) - r_i(T) - r_j(T)\right)$$

If r is 2-SNM-α for deterministic tournaments, then each term in this sum is at most $\mathbb{P}_{\mathcal{T}}(T)\alpha$. It follows that this sum is at most α, and therefore r is also 2-SNM-α for randomized tournaments. ◀

It is straightforward to generalize the above definitions and result to the case of k-SNM-α.

3.4 Other tournament formats

Finally, there are many other tournament formats that are either used in practice or have been previously studied. In this section we show that many of these formats are more susceptible to manipulation than the random single elimination bracket rule; in particular, all of the following formats are at best 2-SNM-1/2.

By far the most common tournament rule for round robin tournaments is some variant of a 'scoring' rule, where the winner is the player who has won the most games (with ties broken in some fashion if multiple players have won the same maximum number of games). In voting theory, this rule is often called Copeland's rule, or Copeland's method [5].

▶ **Definition 16.** A tournament rule r is a *Copeland rule* if the winner is always selected from the set of players with the maximum number of wins.

We begin by showing that no Copeland rule can be 2-SNM-α for any $\alpha < 1$ (regardless of how the rule breaks ties).

▶ **Theorem 17.** *There is no Copeland rule on n players that is 2-SNM-α for $\alpha < 1 - \frac{2}{n-1}$.*

Proof. Assume to begin that $n = 2k + 1$ is odd, and let r be a Copeland rule on n players. Let T be the tournament where each player i beats the k players $\{i + 1, i + 2, \ldots, i + k\}$ but loses to the k players $\{i - 1, i - 2, \ldots, i - k\}$, with indices taken modulo n (similar to the tournament in the proof of Theorem 12).

Since $\sum_{i=1}^{n} r_i(T) = 1$, there must be some i such that $r_{i-1}(T) + r_i(T) \leq \frac{2}{n}$. On the other hand, if player $i - 1$ throws their match to player i, then player i becomes the unique Copeland winner (winning $k + 1$ games) and $r_i(T') = 1$. It follows that, for such a rule, if r is 2-SNM-α, then $\alpha \geq 1 - \frac{2}{n}$.

If n is even, then we can embed the above example for $n - 1$ by assigning one player to be a dummy player that loses to all teams. This immediately implies $\alpha \geq 1 - \frac{2}{n-1}$ in this case. ◀

In [1], Altman and Kleinberg provide three examples of tournament rules that are Condorcet-consistent and monotone: the top cycle rule, the iterative Condorcet rule, and the randomized voting caterpillar rule. We prove lower bounds on α for each of these in turn. Interestingly, the same tournament provides all three lower bounds.

▶ **Definition 18.** The *superman-kryptonite* tournament on n players has i beat j whenever $i < j$, except that player n beats player 1. That is, player 1 beats everyone except for player n, who loses to everyone except for player 1.

Now we show that the superman-kryptonite tournament provides lower bounds against the tournament rules considered in [1].

▶ **Definition 19.** The *top cycle* of a tournament T is the minimal set of players who never lose to any other player. The *top cycle rule* is a tournament rule which assigns the winner to be a uniformly random element of this set.

▶ **Theorem 20.** *The top cycle rule on n players is not 2-SNM-α for any $\alpha < 1 - \frac{2}{n}$.*

Proof. Let T be the superman-kryptonite tournament on n players. The top cycle in T contains all the players, so $r_1(T) + r_n(T) = \frac{2}{n}$. However, if player n throws their match to player 1, player 1 becomes a Condorcet winner and $r_1(T') = 1$. It follows that $\alpha \geq 1 - \frac{2}{n}$. ◀

▶ **Definition 21.** The *iterative Condorcet rule* is a tournament rule that uniformly removes players at random until there is a Condorcet winner, and then assigns that player to be the winner.

▶ **Theorem 22.** *The iterative Condorcet rule on n players is not 2-SNM-α for any $\alpha < \frac{1}{2} - \frac{1}{n(n-1)}$.*

Proof. Let T be the superman-kryptonite tournament on n players. Note that no Condorcet winner will appear until either player 1 is removed, player n is removed, or all other $n - 2$ players are removed. If all the other $n - 2$ players are removed before players 1 or n (which occurs with probability $\frac{2}{n(n-1)}$), then player n wins. If this does not happen and player n is removed before player 1 (which occurs with probability $\frac{1}{2}\left(1 - \frac{2}{n(n-1)}\right) = \frac{1}{2} - \frac{1}{n(n-1)}$), then player 1 becomes the Condorcet winner and wins. Otherwise, player 1 will be removed before player n, while some players in 2 through $n - 1$ remain, and one of them will become the Condorcet winner (the remaining player in $\{2, \ldots, n - 1\}$ with lowest index). It follows that $r_1(T) = \frac{1}{2} - \frac{1}{n(n-1)}$ and $r_n(T) = \frac{2}{n(n-1)}$, so $r_1(T) + r_n(T) = \frac{1}{2} + \frac{1}{n(n-1)}$.

On the other hand, if player n throws their match to player 1, then again player 1 becomes a Condorcet winner and $r_1(T') = 1$. It follows that $\alpha \geq \frac{1}{2} - \frac{1}{n(n-1)}$. ◀

▶ **Definition 23.** The *randomized voting caterpillar rule* is a tournament rule which chooses a winner as follows. Choose a random permutation π of $[n]$. Start by matching $\pi(1)$ and $\pi(2)$, and choose a winner according to T. Then for all $i \geq 3$ match $\pi(i)$ with the winner of the most recent match. The player that wins the last match (against $\pi(n)$) is declared the winner.

▶ **Theorem 24.** *The randomized voting caterpillar rule on n players is not 2-SNM-α for any $\alpha < \frac{1}{2} - \frac{n-3}{n(n-1)}$.*

Proof. Let T be the superman-kryptonite tournament on n players. The only way player 1 loses is if either player n occurs later in π than player 1 (which happens with probability $\frac{1}{2}$) or if $\pi(n) = 1$ and $\pi(1) = 2$ and they play in the first round (which happens with probability $\frac{1}{n(n-1)}$). The only way player n can win is if $\pi(n) = n$ (i.e., they only play the very last game), in which case they will play player 1 and win (this happens with probability $\frac{1}{n}$). It follows that $r_1(T) = \frac{1}{2} - \frac{1}{n(n-1)}$ and $r_n(T) = \frac{1}{n}$, so $r_1(T) + r_n(T) = \frac{1}{2} + \frac{n-2}{n(n-1)}$.

On the other hand, if player n throws their match to player 1, then again player 1 becomes a Condorcet winner and $r_1(T') = 1$. It follows that $\alpha \geq \frac{1}{2} - \frac{n-2}{n(n-1)}$. ◀

References

1. Alon Altman and Robert Kleinberg. Nonmanipulable randomized tournament selections. In Maria Fox and David Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press, 2010. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI10/paper/view/1703.
2. Alon Altman, Ariel D. Procaccia, and Moshe Tennenholtz. Nonmanipulable selections from a tournament. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 27–32, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc. URL: http://dl.acm.org/citation.cfm?id=1661445.1661451.
3. Kenneth J. Arrow. A difficulty in the concept of social welfare. *Journal of Political Economy*, 58(4):328–346, 1950. URL: http://www.jstor.org/stable/1828886.
4. John J. Bartholdi, Craig A. Tovey, and Michael A. Trick. How hard is it to control an election? *Mathematical and Computer Modelling*, 16(8):27 – 40, 1992. doi:http://dx.doi.org/10.1016/0895-7177(92)90085-Y.
5. A.H. Copeland. A 'reasonable' social welfare function. *Seminar on Mathematics in Social Sciences*, 1951.
6. S Cox. Tennis match fixing: Evidence of suspected match-fixing revealed, January 2016. http://www.bbc.com/sport/tennis/35319202.
7. Bhaskar Dutta. Covering sets and a new condorcet choice correspondence. *Journal of Economic Theory*, 44(1):63 – 80, 1988. doi:10.1016/0022-0531(88)90096-8.
8. Peter C. Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33(3):469–489, 1977. doi:10.1137/0133030.
9. Allan Gibbard. Manipulation of voting schemes: a general result. *Econometrica*, 41(4):587–601, 1973.
10. Allan Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica*, 45(3):665–681, 1977. URL: http://www.jstor.org/stable/1911681.
11. P Kelso. Badminton pairs expelled from london 2012 olympics after 'match-fixing' scandal, August 2012. .
12. Michael P. Kim, Warut Suksompong, and Virginia Vassilevska Williams. Who can win a single-elimination tournament? In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 516–522, 2016. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12194.

13 Michael P. Kim and Virginia Vassilevska Williams. Fixing tournaments for kings, chokers, and more. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 561–567, 2015. URL: http://ijcai.org/Abstract/15/085.

14 Jean-Francois Laslier. *Tournament solutions and majority voting*. Number 7 in Studies in Economic Theory. Springer Verlag, 1997.

15 H. Moulin. Choosing from a tournament. *Social Choice and Welfare*, 3(4):271–291, 1986. URL: http://www.jstor.org/stable/41105842.

16 Hervé Moulin, Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

17 Marc Pauly. Can strategizing in round-robin subtournaments be avoided? *Social Choice and Welfare*, 43(1):29–46, 2014. doi:10.1007/s00355-013-0767-6.

18 B Phillips. The tennis triangle, July 2011. http://grantland.com/features/the-tennis-triangle/.

19 Ronald L. Rivest and Emily Shen. An optimal single-winner preferential voting system based on game theory, 2010.

20 Ariel Rubinstein. Ranking the participants in a tournament. *SIAM Journal on Applied Mathematics*, 38(1):108–111, 1980. URL: http://www.jstor.org/stable/2100804.

21 Mark Allen Satterthwaite. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.

22 S Scherer. Italy breaks up soccer match-fixing network involving mafia, May 2015. http://www.bbc.com/news/world-europe-32793892.

23 T. Schwartz. Cyclic tournaments and cooperative majority voting: A solution. *Social Choice and Welfare*, 7(1):19–29, 1990. URL: http://www.jstor.org/stable/41105932.

24 B Sinclair. 12 arrested in esports match fixing scandal - report, October 2015. http://www.gamesindustry.biz/articles/2015-10-19-12-arrested-in-esports-match-fixing-scandal-report.

25 R Smyth. World cup: 25 stunning moments ... no3: West germany 1-0 austria in 1982, February 2014. URL: http://www.theguardian.com/football/blog/2014/feb/25/world-cup-25-stunning-moments-no3-germany-austria-1982-rob-smyth.

26 Isabelle Stanton and Virginia Vassilevska Williams. Rigging tournament brackets for weaker players. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 357–364, 2011. doi:10.5591/978-1-57735-516-8/IJCAI11-069.

27 Thuc Vu, Alon Altman, and Yoav Shoham. On the complexity of schedule control problems for knockout tournaments. In *8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009), Budapest, Hungary, May 10-15, 2009, Volume 1*, pages 225–232, 2009. doi:10.1145/1558013.1558044.

28 H. P. Young. Social choice scoring functions. *SIAM Journal on Applied Mathematics*, 28(4):824–838, 1975. URL: http://www.jstor.org/stable/2100365.

A More Details on our Coupling Argument

In this appendix we present examples of bracket transformations. Recall that our transformations took as input any "bad" bracket, where player i eventually meets player j, *and* player j will lose to some player k in the future if she advances past i (and k is the latest such player). The players benefit from manipulating these brackets. We transformed them into "good" brackets, where either player j is eliminated before even meeting player i, or where

player j would be the champion conditioned on getting past i. The players have no incentive to manipulate these brackets.

We designed two injective transformations with disjoint images, σ_i and σ_j. σ_i was more straight-forward, but we include an example below anyway. σ_j was more complex. We include below an example showing that the complexity is necessary, and then an example of σ_j. All figures are at the end.

A.1 Example of the transformation $\sigma_i(B)$

Recall that σ_i essentially swaps the sub-brackets rooted at i and k. See Section 3.2 for a formal description.

Consider the partial bracket B_1 shown in Figure 6. Then, applying the transformation $\sigma_i(B_1)$ as described in our paper will yield the bracket B_1' shown in Figure 7. Note that this mapping is injective: by examining $\sigma_i(B)$, we see exactly where j is eliminated, and conclude that this must be where i met j in the original B.

A.2 Counterexample to a naive $\sigma_j(B)$

We could try using the same ideas in σ_i for σ_j: simply swap the subtrees rooted at k and j. Unfortunately, this mapping is not injective.

Consider the two brackets B_3, B_4 shown in Figure 8. Then applying this naive transformation will map these brackets to the same bracket (see Figure 9), showing that the mapping may not be injective. This motivates the need for the more involved transformation σ_j from Section 3.2.

Specifically, observe that in B_3, i meets j in round 2, so the depth-2 subtree rooted at k would get swapped with the depth-2 subtree rooted at j. In B_4, i meets j in round 1, so the single node i_1 would get swapped with the single node j. It is easy, but tedious, to complete this into a full tournament/bracket.

A.3 Example of the transformation $\sigma_j(B)$

Essentially, the problem with the naive transformation is that it's hard to recover where i met j in the original B just from the naive $\sigma_j(B)$. This is because maybe on its path to j, i met many other competitors who also would have beaten j, in addition to the k we swap in from the mapping. Our more involved transformation fixes this by additionally swapping all such competitors out of the subtree below i, so we can again recover where i met j in the original B.

Consider the partial bracket B_2 shown in Figure 10 and assume that in the tournament in case i_2 would beat j. Then, applying the transformation $\sigma_j(B_2)$ as described in our paper will yield the bracket B_2' shown in 11.

Note that this mapping is injective! First, we can recover where i met j in the original B by looking at where i first encounters someone who would beat j in $\sigma_j(B)$. Once we learn this, we also know that in the original B, j actually advanced this far in the tournament to meet i, so we know exactly which subtrees we need to un-swap with subtrees of i.

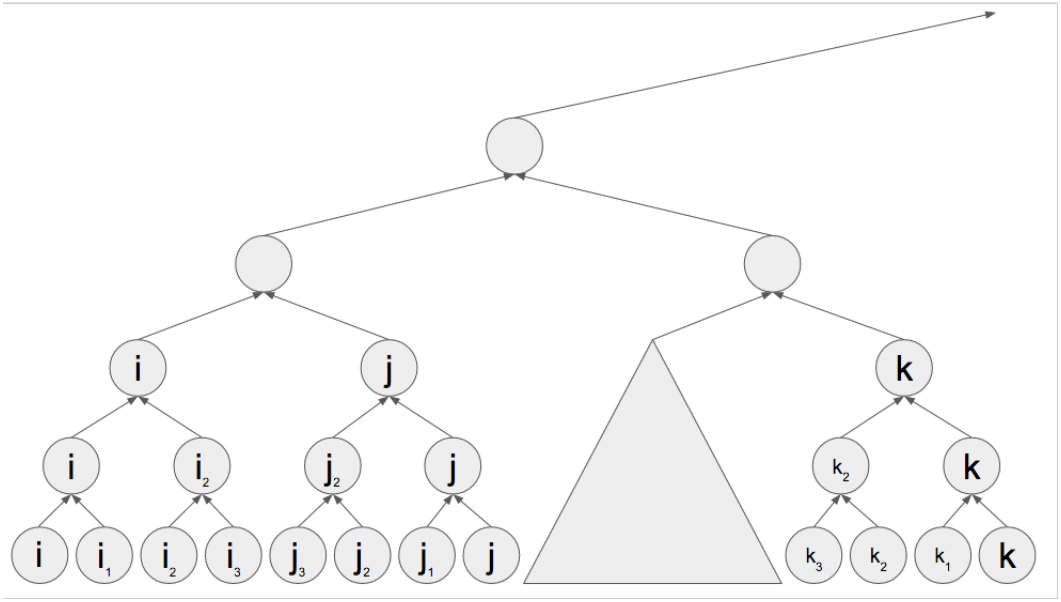


Figure 6 A partial bracket B_1.

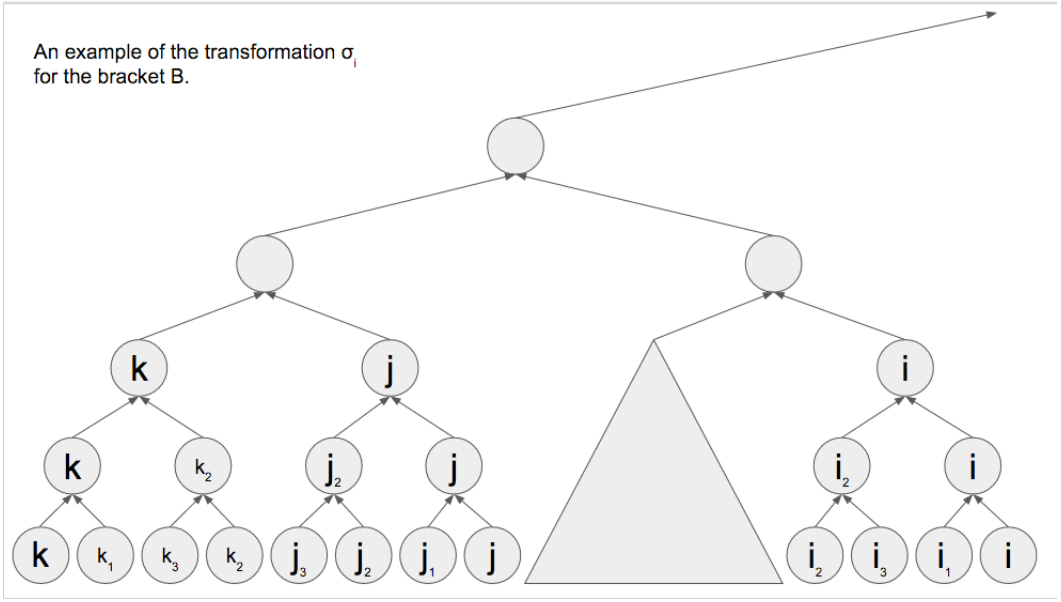


Figure 7 $\sigma_i(B_1)$.

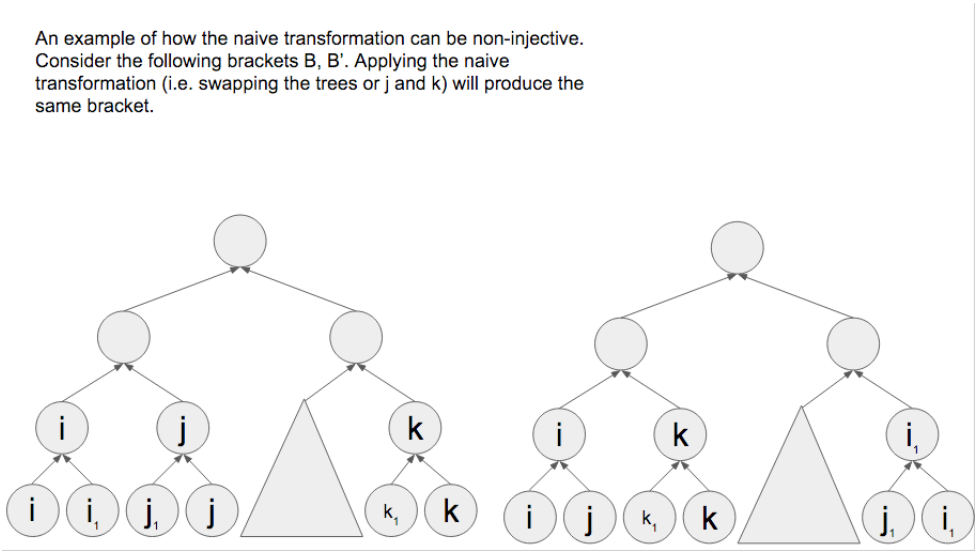


Figure 8 Two partial brackets B_3, B_4.

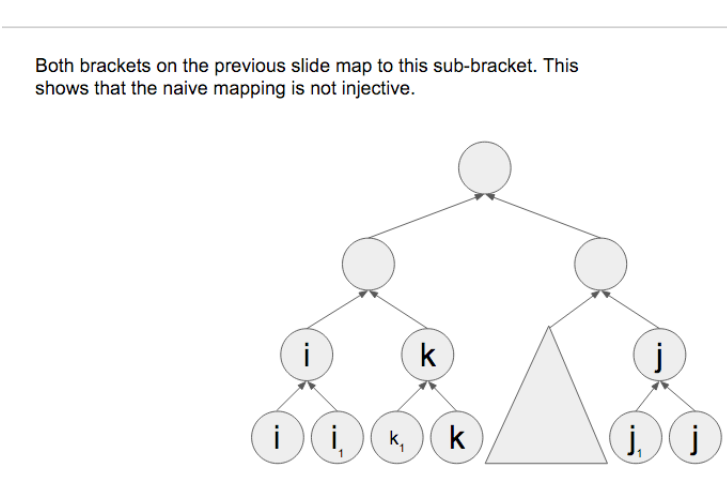


Figure 9 Swapping the subtrees corresponding to j, k in both brackets above yields this bracket.

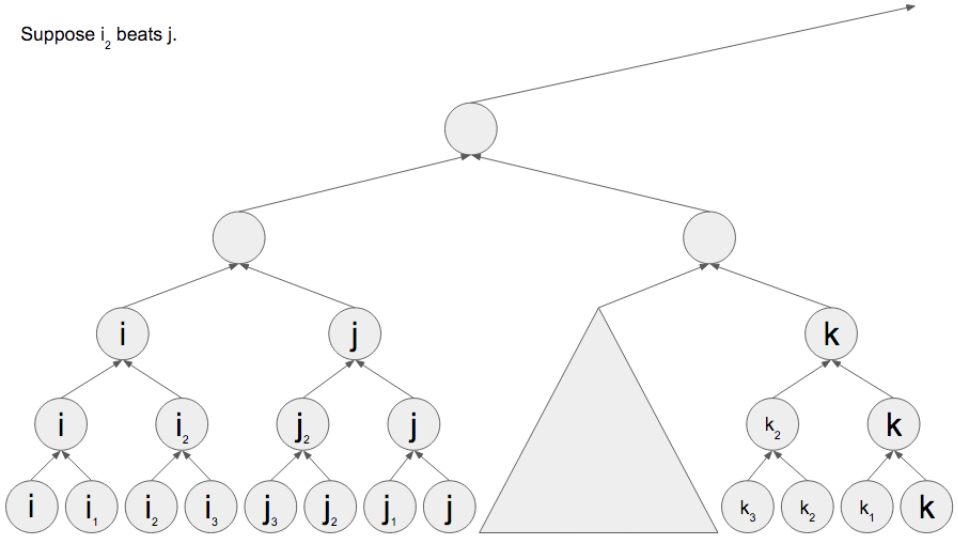


Figure 10 A partial bracket B_2.

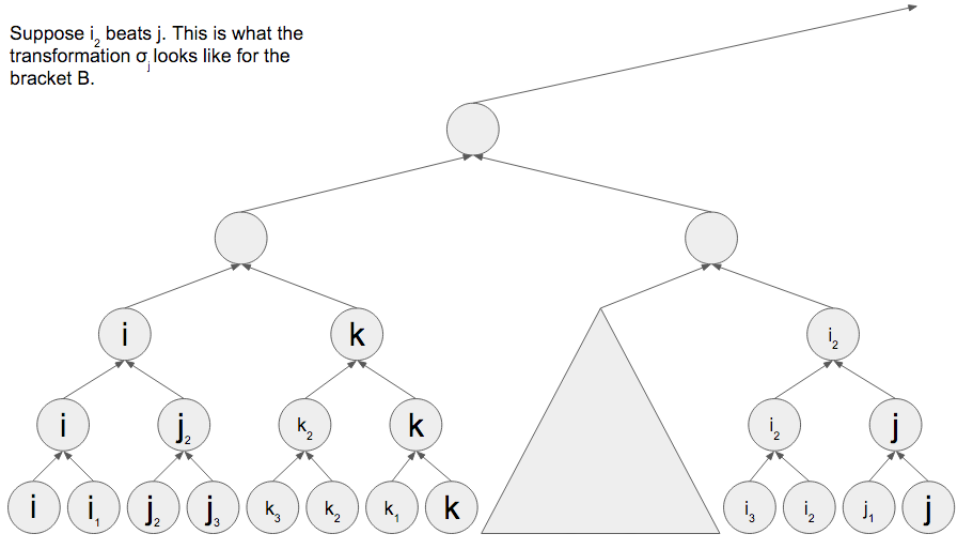


Figure 11 $\sigma_j(B_2)$.

Nash Social Welfare, Matrix Permanent, and Stable Polynomials

Nima Anari*[1], Shayan Oveis Gharan[2], Amin Saberi[3], and Mohit Singh[4]

1 **Stanford University, Stanford, USA**
anari@stanford.edu
2 **University of Washington, Seattle, USA**
shayan@cs.washington.edu
3 **Stanford University, Stanford, USA**
saberi@stanfrod.edu
4 **Microsoft Research, Redmond, USA**
mohitsinghr@gmail.com

Abstract

We study the problem of allocating m items to n agents subject to maximizing the Nash social welfare (NSW) objective. We write a novel convex programming relaxation for this problem, and we show that a simple randomized rounding algorithm gives a $1/e$ approximation factor of the objective, breaking the $1/2e^{1/e}$ approximation factor of Cole and Gkatzelis [8].

Our main technical contribution is an extension of Gurvits's lower bound on the coefficient of the square-free monomial of a degree m-homogeneous stable polynomial on m variables to all homogeneous polynomials. We use this extension to analyze the expected welfare of the allocation returned by our randomized rounding algorithm.

1998 ACM Subject Classification F.2.1 [Numerical Algorithms and Problems] Computations on Polynomials, G.2.1 [Combinatorics] Counting Problems, G.1.6 [Optimization] Convex Programming, G.3 [Probability and Statistics] Probabilistic Algorithms

Keywords and phrases Nash Welfare, Permanent, Matching, Stable Polynomial, Randomized Algorithm, Saddle Point

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.36

1 Introduction

We study the problem of allocating a set of indivisible items to agents subject to maximizing the Nash social welfare (NSW). We are given a set of m indivisible items and we want to assign them to n agents. An allocation vector is a vector $\mathbf{x} \in \{0,1\}^{n\times m}$ such that for each j, exactly one $x_{i,j}$ is 1. We assume that agents have additive valuations. That is, each agent i has nonnegative value $v_{i,j}$ for an item j and the value that i receives for an allocation $\mathbf{x}$ is

$$v_i(\mathbf{x}) = \sum_{j=1}^{m} x_{i,j} v_{i,j}.$$

* This work was partially supported by NSF Award 1216698.


© Nima Anari, Shayan Oveis Gharan, Amin Saberi, and Mohit Singh;
licensed under Creative Commons License CC-BY

The NSW objective is to compute an allocation $\mathbf{x}$ that maximizes the geometric mean of agents' values,

$$\left(\prod_{i=1}^{n} v_i(\mathbf{x})\right)^{\frac{1}{n}}.$$

The above objective naturally encapsulates both fairness and efficiency and has been extensively studied as a notion of fair division (see [14, 5] and references therein).

Recently, there have been a number of results that study the computational complexity of the Nash social welfare objective. For additive valuations it is known that it is NP-hard to approximate the NSW objective within $(1-c)$ [15, 13], for some constant $c > 0$. On the positive side, Nguyen and Rothe [16] designed a $\left(\frac{1}{m-n+1}\right)$ approximation algorithm and Cole and Gkatzelis [8] gave the first constant factor, $\left(\frac{1}{2e^{1/e}}\right)$-approximation. Recently, independent of our work, Cole et al. [7] gave a $\frac{1}{2}$-approximation.

A closely related problem, that captures only fairness, is the Santa-Clause problem where the goal is to find an allocation to maximize the minimum value among all agents, i.e., $\max_{\mathbf{x}} \min_i v_i(\mathbf{x})$ which has also been studied recently [1, 2, 3, 6].

1.1 Our Contributions

Our main contribution is to show an intricate connection between the Nash welfare maximization problem, the theory of real stable polynomials, and the problem of approximating the permanent. We establish this connection in the following manner. We first give a new mathematical programming relaxation for the problem; indeed the standard relaxation has arbitrarily large integrality gap as shown by Cole and Gkatzelis [8]. Our relaxation is a polynomial optimization problem[1] which, despite not being convex in the standard form, can be solved efficiently by a change of variables. We remark that a similar geometric program was used in the context of maximum sub-determinant problem [17].

More precisely, we study a real stable polynomial $p(y_1, \ldots, y_m)$. We give a simple randomized rounding algorithm such that the expected Nash welfare of the allocation returned by the algorithm exactly equals the sum of *square-free* coefficients of $p(\mathbf{y})$. Thus, our program needs to maximize the sum of square-free coefficients of $p(\mathbf{y})$. Unfortunately, such an optimization problem in not convex. Instead, we maximize the following proxy

$$\inf_{\substack{\mathbf{y}>0:\\ \prod_{i\in S} y_i \geq 1, \forall S \in \binom{[m]}{n}}} p(\mathbf{y}).$$

The main part of our analysis is to relate the sum of square-free coefficients of $p(\mathbf{y})$ to the above proxy. This desired inequality is a generalization of an elegant result of Gurvits [11] relating the problem of approximating the permanent of a matrix with the theory of real stable polynomials. We prove this generalization in theorem 1.2. The connection to permanents allows us to use algorithmic results for approximating the permanent due to Jerrum, Sinclair and Vigoda [12] and we obtain the following result.

▶ **Theorem 1.1.** *There is a randomized polynomial time algorithm for the Nash welfare maximization problem that, with high probability, returns a solution with objective at least $1/e$ fraction of the optimum.*

[1] It falls in the broad class of geometric programs, where the mathematical program is convex in logarithms of the variables and not the variables itself.

We emphasize that unlike the recent constant factor approximation algorithms by Cole and Gkatzelis [8] and [7], our algorithm and its analysis are purely algebraic and completely oblivious to the structure of the underlying market. In particular, unlike other approaches we are not taking advantage of the combinatorial structure of "spending restricted assignments" in our rounding algorithms (see [8] for more information). This generality makes our approach potentially applicable to a variety of resource allocation problems.

The crucial ingredient of our analysis is the following general inequality about real stable polynomials that generalizes the result of Gurvits [11] (see theorem 2.5) that provided an elegant proof of the Van-der-Waerden conjecture.

▶ **Theorem 1.2.** *Let p be a degree n-homogeneous real stable polynomial in $y_1, \ldots, y_m$ with nonnegative coefficients. For any set $S \subseteq [m]$, let c_S denote the coefficient of monomial $y^S := \prod_{i \in S} y_i$. If $\sum_{S \in \binom{[m]}{n}} c_S > 0$, then*

$$\begin{aligned} \sum_{S \in \binom{[m]}{n}} c_S &\geq \frac{m! \cdot (m-n)^{m-n}}{m^m \cdot (m-n)!} \inf_{\substack{\mathbf{y}>0: \\ y^S \geq 1, \forall S \in \binom{[m]}{n}}} p(\mathbf{y}) \qquad (1) \\ &\geq e^{-n} \inf_{\substack{\mathbf{y}>0: \\ y^S \geq 1, \forall S \in \binom{[m]}{n}}} p(\mathbf{y}). \end{aligned}$$

Note that second inequality follows by lemma 1.1, $\frac{m!}{m^m} \cdot \frac{(m-k)^{m-k}}{(m-k)!} \geq e^{-k}$. By setting $n = m$ in the above statement, we obtain the result of Gurvits as described in theorem 2.5.

It is not hard to see that the above inequality is (almost) tight. For the stable n-homogeneous polynomial $p(y_1, \ldots, y_m) = (y_1 + \cdots + y_n)^n$, the LHS is $n!$ and the RHS is $(n/e)^n$. This tight example was already studied by Friedland and Gurvits [9] to show tightness of a lower bound for the number of matchings in regular bipartite graphs.

2 Preliminaries

For a vector $\mathbf{y}$, we write $\mathbf{y} \leq 1$ to denote that all coordinates of $\mathbf{y}$ are at most 1. For an integer $n \geq 1$ we use $[n]$ to denote the set of numbers $\{1, 2, \ldots, n\}$. For any m, n, we let $\binom{[m]}{n}$ denote the collection of subsets of $[m]$ of size n.

2.1 Stable Polynomials

Stable polynomials are natural multivariate generalizations of real-rooted univariate polynomials. For a complex number z, let $\mathrm{Im}(z)$ denote the imaginary part of z. We say a polynomial $p(z_1, \ldots, z_m) \in \mathbb{C}[z_1, \ldots, z_m]$ is *stable* if whenever $\mathrm{Im}(z_i) > 0$ for all $1 \leq i \leq m$, $p(z_1, \ldots, z_m) \neq 0$. We say $p(.)$ is real stable, if it is stable and all of its coefficients are real. It is easy to see that any univariate polynomial is real stable if and only if it is real rooted.

We say a polynomial $p(z_1, \ldots, z_m)$ is degree n-homogeneous, or n-homogenous, if every monomial of p has degree exactly n. Equivalently, p is n-homogeneous if for all $a \in \mathbb{R}$, we have

$$p(a \cdot z_1, \ldots, a \cdot z_m) = a^n p(z_1, \ldots, z_m).$$

We say a monomial $z_1^{\alpha_1} \ldots z_m^{\alpha_m}$ is *square-free* if $\alpha_1, \ldots, \alpha_m \in \{0, 1\}$. For a set $S \subset 2^{[m]}$ we write

$$z^S = \prod_{i \in S} z_i.$$

Thus, we can represent a square-free monomial with the set of indices of variables in that monomial.

▶ **Fact 2.1.** *If $p(z_1, \ldots, z_m)$ and $q(z_1, \ldots, z_m)$ are stable then $p \cdot q$ is stable.*

▶ **Fact 2.2.** *The polynomial $\sum_i a_i z_i$ is stable if $a_i \geq 0$ for all i.*

Polynomial optimization problems involving real stable polynomials with nonnegative coefficients can often be turned into concave/convex programs. Such polynomials are log-concave in the positive orthant:

▶ **Theorem 2.3** ([10])**.** *For any n-homogeneous stable polynomial $p(x_1, \ldots, x_n)$ with nonnegative coefficients, $\log p(\mathbf{x})$ is concave in the positive orthant, $\mathbb{R}^n_{++}$.*

It is also an immediate corollary of Hölder's inequality that a polynomial with nonnegative coefficients is log-convex in terms of the log of its variables (for more details on log-convex functions see [4]).

▶ **Fact 2.4.** *For any polynomial $p(y_1, \ldots, y_m)$ with nonnegative coefficients, $\log p(\mathbf{y})$ is convex in terms of $\log \mathbf{y}$. In other words, $\log p(e^{z_1}, \ldots, e^{z_m})$ is convex in terms of $\mathbf{z}$.*

The following theorem is proved by Gurvits [11].

▶ **Theorem 2.5** ([11])**.** *For any degree m-homogeneous stable polynomial $p(z_1, \ldots, z_m)$ with nonnegative coefficients, let $c_{[m]}$ denote the coefficient of the multilinear monomial $z_1 \cdots z_m$. If $c_{[m]} > 0$, then*

$$c_{[m]} \geq \frac{m!}{m^m} \inf_{\mathbf{z}>0} \frac{p(z_1, \ldots, z_m)}{z_1 \ldots z_m}.$$

2.2 Counting Matchings in Bipartite Graphs

Given a bipartite graph $G = (X, Y, E)$ with weights $w : E \to \mathbb{R}$, the weight of a perfect matching M is defined as follows:

$$w(M) = \prod_{e \in M} w_e.$$

Jerrum, Sinclair, and Vigoda in their seminal work designed a FPRAS to count the sum of (weighted) perfect matchings of an arbitrary bipartite graph with nonnegative weights. This problem is also equivalent to the computation of the permanent of a nonnegative matrix.

▶ **Theorem 2.6** ([12])**.** *There exists a randomized polynomial time algorithm that for any arbitrary bipartite graph G with n vertices and nonnegative weights and $\epsilon > 0$ in time polynomial in the size of G and $1/\epsilon$ approximates the sum of weights of all perfect matchings of G within a $1 + \epsilon$ multiplicative error, with high probability.*

A k-matching of a bipartite graph $G = (X, Y, E)$ is a set $M \subseteq E$ of size $|M| = k$ such that no two edges share an endpoint. The following corollary follows immediately from the above theorem. For completeness, we prove it in the appendix.

▶ **Corollary 2.7.** *There is a randomized polynomial time algorithm that for any arbitrary bipartite graph G with nonnegative edge weights and for any given $\epsilon > 0$ and integer $k \leq n$ in time polynomial in the size of G and $1/\epsilon$ approximates the sum of the weights of all k-matchings of G within $1 + \epsilon$ multiplicative error, with high probability.*

3 Approximation Algorithm for NSW Maximization

In this section, we give an approximation algorithm for the NSW maximization problem. We begin by giving a mathematical programming relaxation that can be efficiently solved. For convenience, we will aim to optimize

$$\left(\prod_{i=1}^{n} v_i(\mathbf{x})\right),$$

which is the n^{th} power of the NSW objective. Thus, it is enough to give an e^{-n}-approximation to the above objective. With a slight abuse of notation, we will also refer to problem of maximizing the new objective as the Nash welfare problem. In section 3.2, we give a rounding algorithm that proves the guarantee claimed in Theorem 1.1.

3.1 Mathematical Programming Relaxation

We use the following mathematical program.

$$\begin{aligned} \max_{\mathbf{x}} \quad & \inf_{\mathbf{y}>0: y^S \geq 1, \forall S \in \binom{[m]}{n}} \quad \prod_{i=1}^{n}\left(\sum_{j=1}^{m} x_{i,j} v_{i,j} y_j\right), \\ \text{s.t.} \quad & \sum_{i=1}^{n} x_{i,j} \leq 1 \qquad \forall 1 \leq j \leq m, \\ & x_{i,j} \geq 0 \qquad \forall i,j. \end{aligned} \tag{2}$$

First, we show that (2) is a relaxation of the Nash welfare problem and can be optimized in polynomial time to an arbitrary accuracy.

▶ **Lemma 3.1.** *The mathematical program* (2) *is a relaxation of the Nash welfare problem and can be optimized in polynomial time.*

Proof. Let $x^* \in \{0,1\}^{n\times m}$ be an optimal solution of the Nash welfare problem and let $\sigma : [m] \to [n]$ denote the assignment, i.e., $\sigma(j) = i$ if and only if $x^*_{ij} = 1$. We show that x^* is a feasible solution (2) of objective $\prod_{i=1}^{n} v_i(x^*)$. Consider any $\mathbf{y} > 0$ such that $y^S \geq 1$ for each $S \subseteq \binom{[m]}{n}$. Moreover let $\mathcal{S} = \{S \in \binom{[m]}{n} : \forall i \in [n], \exists j \in S \text{ such that } x^*_{ij} = 1\}$. We have

$$\begin{aligned} \prod_{i=1}^{n}\left(\sum_{j=1}^{m} x^*_{i,j} v_{i,j} y_j\right) &= \textstyle\sum_{S\in\mathcal{S}} y^S \prod_{j\in S} v_{\sigma(j),j} \\ &\geq \textstyle\sum_{S\in\mathcal{S}} \prod_{j\in S} v_{\sigma(j),j} \\ &= \textstyle\prod_{i=1}^{n}\left(\sum_{j=1}^{m} x^*_{i,j} v_{i,j}\right) \end{aligned}$$

as required, where we use the fact that $y^S \geq 1$ for each $S \in \mathcal{S}$. To show that the objective of the mathematical program equals $\prod_{i=1}^{n} v_i(x^*)$, we consider the solution $y^*_j = 1$ for each $j \in [m]$.

To solve the mathematical program, we observe that the function $\log \prod_{i=1}^{n} \sum_{j=1}^{m} x_{i,j} v_{i,j} y_j$ is concave in x and convex in $\log \mathbf{y}$, where $\log \mathbf{y}$ is the vector defined by taking logarithms of the vector $\mathbf{y}$ coordinate-wise. These follow from theorem 2.3 and fact 2.4. Moreover, the constraints on $\mathbf{x}$ and $\log \mathbf{y}$ are linear. Thus the above program can be formulated as a convex program and solved to an arbitrary accuracy. ◀

Algorithm 1 An Algorithm for NSW Maximization.

Check whether the optimal solution has weight strictly more than zero using the bipartite matching algorithm. Return zero if answer is false.
Find an optimal solution $\mathbf{x}^*$ to the mathematical program (2).
Independently for each item $j \in [m]$, assign item j to one agent where agent $i \in [n]$ is chosen with probability x^*_{ij}.

3.2 Randomized Algorithm I

We now give a rounding algorithm that proves the required guarantee. Algorithm 1 will only satisfy the guarantee in expectation. Later, we show how to give a randomized algorithm that gives essentially the same guarantee with high probability.

The first step of the algorithm can be implemented by a bipartite matching problem. Indeed consider the bipartite graph with one side as agents and other as items. We have an edge (i, j) for agent i and item j if $v_{ij} > 0$. The optimal solution to the NSW maximization problem is strictly positive if and only if this bipartite graph has a matching that includes an edge at every agent. Thus, we can check in polynomial time whether the optimal solution has weight zero. For the remainder of the section, we assume that the optimal solution is strictly positive.

We say $\mathbf{x} \in \mathbb{R}_+^{n\times m}$ is a fractional allocation vector if for each $j \in [m]$, $\sum_{i=1}^n x_{i,j} = 1$. Given any fractional allocation $\mathbf{x}$, consider the following polynomial in variables $y_1, \ldots, y_n$,

$$p_{\mathbf{x}}(y_1, \ldots, y_n) = \prod_{i=1}^{n} \left(\sum_{j=1}^{m} x_{i,j} v_{i,j} y_j \right).$$

Observe that $p_{\mathbf{x}}(\mathbf{y})$ is a degree n-homogenous polynomial in m variables for any $\mathbf{x}$ or the identically 0 polynomial.

▶ **Lemma 3.2.** *We have the following.*

1. *For $S \subseteq [m]$ of size n, let c_S denote the coefficient of y^S in $p_{\mathbf{x}^*}(\mathbf{y})$. Then, the expected value of algorithm 1 equals*

$$\sum_{S \in \binom{[m]}{n}} c_S.$$

2. *The optimal value of the relaxation (2) is*

$$\inf_{\mathbf{y}: y^S \geq 1, \forall S \in \binom{[m]}{n}} p_{\mathbf{x}^*}(\mathbf{y}).$$

Proof. Let $X_{i,j}$ be the random variable indicating that j is assigned to i. Then, the value that i receives is $\sum_{j=1}^m X_{i,j} v_{i,j}$. So, the expected value of the algorithm is

$$\mathbb{E}\left[\prod_{i=1}^{n}\sum_{j=1}^{m} X_{i,j} v_{i,j}\right] = \sum_{\sigma:[n]\to[m]} \mathbb{E}\left[\prod_{i=1}^{n} X_{i,\sigma(i)} v_{i,\sigma(i)}\right] =$$

$$\sum_{\sigma:[n]\to[m]} \mathbb{P}\left[\forall i : X_{i,\sigma(i)} = 1\right] \prod_{i=1}^{n} v_{i,\sigma(i)}.$$

where σ is summed over all functions from $[n]$ to $[m]$. Observe that $\mathbb{P}\left[\forall i : X_{i,\sigma(i)=1}\right] \neq 0$ only if σ is a one-to-one function. In such a case, we have $\mathbb{P}\left[\forall i : X_{i,\sigma(i)} = 1\right] = \prod_{i=1}^{n} x_{i,\sigma(i)}$ where we use the fact that each item is assigned independently. Therefore,

$$\mathbb{E}\left[\prod_{i=1}^{n}\sum_{j=1}^{m} X_{i,j}v_{i,j}\right] = \sum_{\sigma:[n]\underset{\text{one-to-one}}{\to}[m]}\prod_{i=1}^{n} x_{i,\sigma(i)}v_{i,\sigma(i)}.$$

The lemma follows by the fact that for any one-to-one σ, the term $\prod_{i=1}^{n} x_{i,\sigma(i)}v_{i,\sigma(i)}$ on the RHS appears in the coefficient of the (square-free) monomial $\prod_{i=1}^{n} y_{\sigma(i)}$ of $p_{\mathbf{x}^*}(\mathbf{y})$. For any $S \in \binom{[m]}{n}$ the coefficient of y^S in $p_{\mathbf{x}^*}(\mathbf{y})$ is the sum of all such terms where $\sigma([n]) = S$.

The proof of the second claim is immediate by definition. ◀

We are now ready to apply theorem 1.2 and obtain the following immediate corollary.

▶ **Corollary 3.3.** *The expected objective of algorithm 1 is at least*

$$e^{-n} \cdot \mathrm{OPT}$$

where OPT *is the optimal NSW objective.*

Proof. From fact 2.1 and fact 2.2, it follows that $p_{\mathbf{x}^*}(\mathbf{y})$ as defined above is real stable with nonnegative coefficients. Moreover, it is an m-variate polynomial that is degree n-homogenous. Let c_S denote the coefficient of square-free monomial y^S for any $S \in \binom{[m]}{n}$. Since, we assume that there is at least one assignment that has strictly positive NSW objective, the sum of coefficients $\sum_{S\in\binom{[m]}{n}} c_S > 0$. Thus, from theorem 1.2, we have

$$\sum_{S\in\binom{[m]}{n}} c_S \quad \geq \quad e^{-n} \min_{\mathbf{y}:y^S\geq 1,\forall S\in\binom{[m]}{n}} p_{\mathbf{x}^*}(\mathbf{y}).$$

Now the proof is immediate using lemma 3.2. ◀

3.3 Randomized Algorithm II

From corollary 3.3, the expected NSW of the allocation returned by algorithm 1 is at least $1/e^n$ fraction of the optimum. Repeated applications of the algorithm to obtain a high probability bound is not possible since the output of algorithm 1 may have an exponentially large variance. In this section, we prove Theorem 1.1 by giving an algorithm that returns the same guarantee as algorithm 1 with high probability.

Proof of theorem 1.1. We use the method of conditional expectations to prove the theorem. We iteratively assign one item at a time, making sure that conditional expectation over the random assignment of the remaining items does not decrease (substantially). We now claim that for any assignment x, the expected value of the objective as given by randomized algorithm 1 equals the number of weighted n-matchings of a bipartite graph. Consider the weighted bipartite graph $G = ([n], [m], E)$ where for any $1 \leq i \leq n$ and $1 \leq j \leq m$, $w_{i,j} = x_{i,j}v_{i,j}$. Then, for one-to-one mapping $\sigma : [n] \to [m]$, the coefficient of the monomial $\prod_{i=1}^{n} x_{i,\sigma(i)}v_{i,\sigma(i)}$ is equal to the weight of the n-matching $\{(1,\sigma(1)),(2,\sigma(2)),\ldots,(n,\sigma(n))\}$. Therefore, the sum of square-free monomials of $p_{\mathbf{x}}(\mathbf{y})$ is equal to the sum of the weights of all n-matchings of G.

Now, pick any item $j \in [m]$ and any fractional assignment x. Consider the following n assignments, $x^1, \ldots, x^n$. Assignment x^i assigns item j to i and rest of the items identically

to the fractional assignment x. Thus $x^i_{ij} = 1$, $x^i_{i',j} = 0$ for all $i \neq i'$ and $x^i_{i'j'} = x_{i'j'}$ for any $j' \neq j$. Let ALG^i denote the objective value of the output of algorithm 1 on solution x^i and ALG on x. Since the objective value of the algorithm 1 is linear in $\{x_{ij} : i \in [n]\}$ for fixed j, we have

$$\mathrm{ALG} = \sum_{i=1}^{n} x_{ij}\, \mathrm{ALG}^i$$

Thus ALG is the expected value of the conditional expected value of the output of the algorithm 1 when we assign item j to one of the agents; it is assigned to agent i with probability x_{ij}.

By corollary 2.7, we can estimate ALG and ALG^i within a factor of $1 + 1/m^3$ factor in polynomial time. Therefore, using the method of conditional expectations, we obtain an allocation of NSW of value at least $\frac{\mathrm{OPT}}{e^n} \cdot (1 - 1/m^3)^m \geq \frac{\mathrm{OPT}}{\left((1+\frac{1}{n})e\right)^n}$ where OPT denotes the objective of the optimal allocation. ◀

4 A Generalization of Gurvits's Theorem

In this section we prove theorem 1.2. Let

$$q(y_1, \ldots, y_m) = (y_1 + \cdots + y_m)^{m-n}$$

be a degree $(m-n)$-homogenous polynomial. It is straightforward to see that it is real stable. Consider the polynomial $p(\mathbf{y})q(\mathbf{y})$. Observe that this is a degree m-homogeneous stable polynomial with nonnegative coefficients. Since from the assumption of theorem 1.2, at least one of the square-free monomials in $p(\mathbf{y})$ has a non-zero coefficient, the coefficient of the square-free monomial in $p(\mathbf{y})q(\mathbf{y})$ is non-zero. Let $\alpha_{[m]}$ be the coefficient of the square-free monomial $y_1 \cdots y_m$ in $p(\mathbf{y})q(\mathbf{y})$. Thus, from theorem 2.5, we have

$$\alpha_{[m]} \geq \frac{m!}{m^m} \inf_{\mathbf{y}>0} \frac{p(\mathbf{y})q(\mathbf{y})}{y_1 \ldots y_m}. \tag{3}$$

To prove theorem 1.2 it is enough to relate the LHS and the RHS of (3) to the two sides of (1). This is done in lemma 4.1 and proposition 4.2.

▶ **Lemma 4.1.** *We have*

$$(m-n)! \sum_{S \in \binom{[m]}{n}} c_S = \alpha_{[m]}.$$

Proof. The RHS is the coefficient of the square-free monomial $y_1 \ldots, y_m$ in $p(\mathbf{y})q(\mathbf{y})$. The square-free monomial of $p(\mathbf{y})q(\mathbf{y})$ is obtained whenever we multiply a square-free monomial y^S of $p(\mathbf{y})$ with the square-free monomial $y^{\overline{S}}$ of $q(\mathbf{y})$ for some $S \in \binom{[m]}{n}$. Lemma's statement follows by the fact that the coefficient of $y^{\overline{S}}$ in $q(\mathbf{y})$ is $(m-n)!$ for every $S \in \binom{[m]}{n}$ and the coefficient of y^S in $p(\mathbf{y})$ is c_S. ◀

The proof of theorem 1.2 is now immediate from the following proposition which relates the RHS of (3) and (1).

▶ **Proposition 4.2.**

$$\inf_{\mathbf{y}>0} \frac{p(\mathbf{y})q(\mathbf{y})}{y_1 \ldots y_m} \geq (m-n)^{m-n} \inf_{\mathbf{y}>0 : y^S \geq 1, \forall S \in \binom{[m]}{n}} p(\mathbf{y}).$$

In the rest of this section we prove the above proposition. We do the proof in two steps. First, we use convex duality to simplify the RHS, and then we prove the proposition.

▶ **Lemma 4.3.**

$$\inf_{\mathbf{y}>0: y^S \geq 1, \forall S \in \binom{[m]}{n}} p(\mathbf{y}) = \sup_{0\leq\theta\leq 1: \sum_{j=1}^m \theta_j = n} \inf_{\mathbf{y}>0} \frac{p(\mathbf{y})}{y_1^{\theta_1}\dots y_m^{\theta_m}}.$$

Proof. The proof follows by convex duality. By taking logarithm of $p(\mathbf{y})$ and the change of variable $z_j = e^{y_j}$, we obtain the following equivalent convex program to the LHS of the above inequality.

$$\begin{aligned} \inf \quad & \log p(e^{z_1}, \dots, e^{z_m}) \\ \text{s.t.} \quad & \sum_{i\in S} z_i \geq 0 \qquad\qquad \forall S \in \binom{[m]}{n}. \end{aligned} \tag{4}$$

Let λ_S be the Lagrange dual variable associated to the constraint corresponding to the set $S \in \binom{[m]}{n}$. The Lagrangian of the above convex program is defined as follows:

$$L(\mathbf{z}, \lambda) = \log p(e^{z_1}, \dots, e^{z_m}) - \sum_{S\in\binom{[m]}{n}} \lambda_S \sum_{i\in S} z_i.$$

The Lagrange dual to (4) is

$$\sup_{\lambda\geq 0} \inf_{\mathbf{z}} L(\mathbf{z}, \lambda).$$

Since $p(\mathbf{y})$ has a non-zero coefficient for at least one of the square-free monomials, the objective of the convex program (4) is finite for any $\mathbf{z}$ and it is easy to see that the Slater conditions are satisfied. Thus the optimum value of the Lagrange dual is exactly equal to the optimum of (4).

Let $\mathbf{z}^*, \lambda^*$ be an optimum of the above program. We claim that $\sum_S \lambda_S^* = 1$. This simply follows from first order optimality conditions. If $\sum_S \lambda_S^* < 1$, then

$$\begin{aligned} L(\mathbf{z}^* - \epsilon, \lambda^*) &= \log p(e^{z_1^*-\epsilon}, \dots, e^{z_m^*-\epsilon}) - \sum_{S\in\binom{[m]}{n}} \lambda_S^* \sum_{j\in S} (z_j^* - \epsilon) \\ &= L(\mathbf{z}^*, \lambda^*) - n\cdot\epsilon + \sum_{S\in\binom{[m]}{n}} n\lambda_S^*\epsilon < L(\mathbf{z}^*, \lambda^*). \end{aligned}$$

Similarly, if $\sum_{S\in\binom{[m]}{n}} \lambda_S > 1$, $L(\mathbf{z}^* + \epsilon, \lambda^*) < L(\mathbf{z}^*, \lambda^*)$. So, λ^* is a probability distribution on sets of size n. We let $L'(\mathbf{z}, \theta) = \log p(e^{\mathbf{z}}) - \sum_{j=1}^m z_j\theta_j$ Thus, we obtain that

$$\sup_{0\leq\theta\leq 1: \sum_{j=1}^m \theta_j = n} \inf_{\mathbf{z}} L'(\mathbf{z}, \theta) \geq \sup_{\lambda\geq 0} \inf_{\mathbf{z}} L(\mathbf{z}, \lambda).$$

by setting $\theta_j^* = \sum_{S\in\binom{[m]}{n}: j\in S} \lambda_S^*$ to be the marginal probability of the element j.

We now claim that equality must hold in the above. This follows since given any $\theta \in \{0 \leq \theta \leq 1 : \sum_{j=1}^m \theta_j = n\}$, there exists a probability distribution over sets of size n such that marginal of every element is exactly θ_j. Setting λ'_S to be the probability of set $S \in \binom{[m]}{n}$, we obtain that for any $\mathbf{z}$ and θ, we have $L'(\mathbf{z}, \theta) = L(\mathbf{z}, \lambda')$. Putting this together we have

$$\inf_{\sum_{j\in S} z_j \geq 0, \forall S\in\binom{[m]}{n}} \log p(e^{\mathbf{z}}) = \sup_{0\leq\theta\leq 1: \sum_{j=1}^m \theta_j = n} \inf_{\mathbf{z}} \left(\log p(e^{\mathbf{z}}) - \sum_{j=1}^m z_j\theta_j \right).$$

Substituting e^{z_j} with y_j and taking the exponential of the objective functions we have

$$\inf_{\mathbf{y}>0: y^S \geq 1, \forall S \in \binom{[m]}{n}} p(\mathbf{y}) = \sup_{0 \leq \theta \leq 1: \sum_{j=1}^m \theta_j = n} \inf_{\mathbf{y}>0} \frac{p(\mathbf{y})}{y_1^{\theta_1} \cdots y_m^{\theta_m}}$$

as desired. ◀

Now we give the proof of proposition 4.2.

Proof of proposition 4.2. By lemma 4.3, it is enough to show that

$$\inf_{\mathbf{y}>0} \frac{p(\mathbf{y})q(\mathbf{y})}{y_1 \dots y_m} \geq (m-n)^{m-n} \sup_{0 \leq \theta \leq 1: \sum_{j=1}^m \theta_j = n} \inf_{\mathbf{y}>0} \frac{p(\mathbf{y})}{y_1^{\theta_1} \cdots y_m^{\theta_m}}.$$

Let θ be any vector such that $0 \leq \theta \leq 1$ and $\sum_i \theta_i = n$. It is enough to show for any such θ,

$$\inf_{\mathbf{y}>0} \frac{p(\mathbf{y})q(\mathbf{y})}{y_1 \dots y_m} \geq (m-n)^{m-n} \inf_{\mathbf{y}>0} \frac{p(\mathbf{y})}{y_1^{\theta_1} \cdots y_m^{\theta_m}}.$$

We prove a stronger statement,

$$\inf_{\mathbf{y}>0} \frac{p(\mathbf{y})}{y_1^{\theta_1} \dots y_m^{\theta_m}} \cdot \inf_{\mathbf{y}>0} \frac{q(\mathbf{y})}{y_1^{1-\theta_1} \cdot y_m^{1-\theta_m}} \geq (m-n)^{m-n} \inf_{\mathbf{y}>0} \frac{p(\mathbf{y})}{y_1^{\theta_1} \dots y_m^{\theta_m}}.$$

Equivalently, we show that

$$\inf_{\mathbf{y}>0} \frac{q(\mathbf{y})}{y_1^{1-\theta_1} \dots y_m^{1-\theta_m}} \geq (m-n)^{m-n}$$

Taking $(m-n)$-th root of both sides it is enough to show that

$$\inf_{\mathbf{y}>0} \frac{y_1 + \dots + y_m}{y_1^{\alpha_1} \dots y_m^{\alpha_m}} \geq m-n, \tag{5}$$

where $\alpha_j = \frac{1-\theta_j}{m-n}$ for all $j \in [m]$. Note that by the definition of θ, we have $0 \leq \alpha_j \leq \frac{1}{m-n}$ and that

$$\sum_i \alpha_j = \frac{m - \sum_{j=1}^m \theta_j}{m-n} = 1.$$

Therefore, the ratio on the LHS of (5) is homogeneous in $\mathbf{y}$. Thus, to prove (5), it is enough to prove the following

$$\sup_{\mathbf{y}>0: \sum_{j=1}^m y_j = 1} y_1^{\alpha_1} \dots y_m^{\alpha_m} \leq \frac{1}{m-n}. \tag{6}$$

Next, we use the weighted AM-GM inequality. We let $\alpha_1, \dots, \alpha_m$ be the weights, and recall that α_j's sum to 1. Weighted AM-GM implies that

$$\sum_{j=1}^m \alpha_j \frac{y_j}{\alpha_j} \geq \prod_{j=1}^m \left(\frac{y_j}{\alpha_j}\right)^{\alpha_j} = \prod_{j=1}^m \alpha_j^{-\alpha_j} \prod_{j=1}^m y_j^{\alpha_j}$$

Therefore,

$$\sup_{\mathbf{y}>0: \sum_{j=1}^m y_j = 1} \prod_{j=1}^m y_j^{\alpha_j} \leq \prod_{j=1}^m \alpha_j^{\alpha_j}.$$

To prove (6), it is enough to show that

$$\prod_{j=1}^{m} \alpha_j^{\alpha_j} \leq \frac{1}{m-n}.$$

Or equivalently,

$$\sum_{j=1}^{m} -\alpha_j \log \alpha_j \geq \log(m-n).$$

Since $\alpha_j \leq \frac{1}{m-n}$ and that $\sum_{j=1}^{m} \alpha_j = 1$, we have

$$\sum_{j=1}^{m} -\alpha_j \log \alpha_j \quad \geq \quad \sum_{j=1}^{m} -\alpha_j \log \frac{1}{m-n} = \log(m-n) \sum_{j=1}^{m} \alpha_j = \log(m-n),$$

as required. ◀

Acknowledgements. The authors would like to thank Leonid Gurvits for insightful comments and pointers to the literature.

References

1. Arash Asadpour, Uriel Feige, and Amin Saberi. Santa claus meets hypergraph matchings. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 10–20. Springer, 2008.
2. Arash Asadpour and Amin Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. In *STOC*, pages 114–121, 2007.
3. N Bansal and M Sviridenko. The santa claus problem. In *Proceedings of Symposium on Theory of Computing*, pages 31–40, 2006.
4. S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2006.
5. Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum nash welfare. In *EC*, 2016.
6. Deeparnab Chakrabarty, Julia Chuzhoy, and Sanjeev Khanna. On allocating goods to maximize fairness. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 107–116. IEEE, 2009.
7. Richard Cole, Nikhil R. Devanur, Vasilis Gkatzelis, Kamal Jain, Tung Mai, Vijay V. Vazirani, and Sadra Yazdanbod. Convex program duality, fisher markets, and nash social welfare. submitted, 2016.
8. Richard Cole and Vasilis Gkatzelis. Approximating the nash social welfare with indivisible items. In *STOC*, pages 371–380. ACM, 2015.
9. Shmuel Friedland and Leonid Gurvits. Lower bounds for partial matchings in regular bipartite graphs and applications to the monomer–dimer entropy. *Combinatorics, Probability and Computing*, 17(03):347–361, 2008.
10. Osman Güler. Hyperbolic polynomials and interior point methods for convex programming. *MOR*, 22(2):350–77, 1997.
11. Leonid Gurvits. Hyperbolic polynomials approach to van der waerden/schrijver-valiant like conjectures: Sharper bounds, simpler proofs and algorithmic applications. In *STOC*, STOC '06, pages 417–426, 2006.
12. Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51(4):671–697, July 2004.

13 Euiwoong Lee. Apx-hardness of maximizing nash social welfare with indivisible items. arXiv preprint arXiv:1507.01159, 2015.

14 Hervé Moulin. *Fair division and collective welfare*. MIT press, 2004.

15 Nhan-Tam Nguyen, Trung Thanh Nguyen, Magnus Roos, and Jörg Rothe. Computational complexity and approximability of social welfare optimization in multiagent resource allocation. *Autonomous agents and multi-agent systems*, 28(2):256–289, 2014.

16 Trung Thanh Nguyen and Jörg Rothe. Minimizing envy and maximizing average nash social welfare in the allocation of indivisible goods. *Discrete Applied Mathematics*, 179:54–68, 2014.

17 Aleksandar Nikolov and Mohit Singh. Maximizing determinants under partition constraints. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 192–201, 2016.

A Miscellaneous Lemmas

▶ **Lemma 1.1.** *For any $k \leq m$, we have*

$$\frac{m!}{m^m} \cdot \frac{(m-k)^{m-k}}{(m-k)!} \geq e^{-k}.$$

Proof. We prove by induction on k. The claim obviously holds for $k = 0$. For the induction step, it is sufficient to show that

$$\frac{1}{e} \cdot \frac{m!}{m^m} \cdot \frac{(m-k)^{m-k}}{(m-k)!} \leq \frac{m!}{m^m} \cdot \frac{(m-(k+1))^{m-(k+1)}}{(m-(k+1))!}.$$

Equivalently, it is enough to show that

$$\left(\frac{m-k}{m-(k+1)}\right)^{m-(k+1)} \leq e.$$

The above can be written as $(1 + \frac{1}{m-k-1})^{m-k-1} \leq e$. The latter follows by the fact that $1 + x \leq e^x$. ◀

Proof of corollary 2.7. Suppose that we are given a bipartite graph $G = (X, Y, E)$ where $X = \{x_1, \ldots, x_m\}$ and $Y = \{y_1, \ldots, y_n\}$. Note that m is not necessarily equal to n. We construct another graph $G' = (X', Y', E')$ such there is a one-to-$(m-k)!(n-k)!$ and onto mapping between the k-matchings of G and the perfect matchings of G'. That is each k-matching of G is mapped to a unique set of $(m-k)!(n-k)!$ perfect matchings of G', and for each perfect matching M' of G' there is a k-matching of G that has M' in its image.

Let $X' = X \cup \{x_{m+1}, \ldots, x_{m+n-k}\}$ and $Y' = Y \cup \{y_{n+1}, \ldots, y_{m+n-k}\}$. The set of edges E' is the union of E and the following edges: Connect all vertices of $X' \setminus X$ to all vertices of Y with weight 1, and connect all vertices of $Y' \setminus Y$ to all vertices of X with weight 1. Observe that for any k-matching M of G there are exactly $(m-k)!(n-k)!$ perfect matchings in G' that contain M; for any such prefect matching M', we have $M' \setminus M \subseteq E' \setminus E$. So, this mapping is one-to-$(m-k)!(n-k)!$. Furthermore, any perfect matching M' of G' has exactly k edges in E, i.e., $|M' \cap E| = k$. So, this mapping is onto.

It follows that a $1+\epsilon$ approximation to the sum of the weights of all perfect matchings of G' is a $1+\epsilon$ approximation to the sum of the weights of all k-matchings of G. ◀

Multiplayer Parallel Repetition for Expanding Games

Irit Dinu*[1], Prahladh Harsha†[2], Rakesh Venkat[3], and Henry Yuen[4]

1 **Weizmann Institute of Science, Rehovot, Israel**
irit.dinur@weizmann.ac.il
2 **Tata Institute of Fundamental Research, Mumbai, India**
prahladh@tifr.res.in
3 **Tata Institute of Fundamental Research, Mumbai, India**
rakesh09@gmail.com
4 **University of California at Berkeley, Berkeley, USA**
hyuen@cs.berkeley.edu

Abstract

We investigate the value of parallel repetition of one-round games with any number of players $k \geq 2$. It has been an open question whether an analogue of Raz's Parallel Repetition Theorem holds for games with more than two players, i.e., whether the value of the repeated game decays exponentially with the number of repetitions. Verbitsky has shown, via a reduction to the density Hales-Jewett theorem, that the value of the repeated game must approach zero, as the number of repetitions increases. However, the rate of decay obtained in this way is extremely slow, and it is an open question whether the true rate is exponential as is the case for all two-player games.

Exponential decay bounds are known for several special cases of multi-player games, e.g., free games and anchored games. In this work, we identify a certain expansion property of the base game and show all games with this property satisfy an exponential decay parallel repetition bound. Free games and anchored games satisfy this expansion property, and thus our parallel repetition theorem reproduces all earlier exponential-decay bounds for multiplayer games. More generally, our parallel repetition bound applies to all multiplayer games that are *connected* in a certain sense.

We also describe a very simple game, called the GHZ game, that does *not* satisfy this connectivity property, and for which we do not know an exponential decay bound. We suspect that progress on bounding the value of this the parallel repetition of the GHZ game will lead to further progress on the general question.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes

Keywords and phrases Parallel Repetition, Multi-player, Expanders

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.37

1 Introduction and Results

We consider multi-player one-round games, and their parallel repetition. In a k-player game G, a referee chooses a k-tuple of questions $(x^1, \ldots, x^k)$ from some question distribution μ, and sends x^t to player t. Each player t gives an answer a^t that only depends on their question (i.e., they cannot communicate with each other). The referee evaluates the players' questions

* Research of the author supported in part by an ISF-UGC grant number 1399/14.
† Research of the author supported in part by an ISF-UGC grant number 1399/14.


© Irit Dinur, Prahladh Harsha, Rakesh Venkat, and Henry Yuen;
licensed under Creative Commons License CC-BY

and answers according to some predicate $V((x^1, \ldots, x^k), (a^1, \ldots, a^k))$, and the players win if this predicate evaluates to 1. The *value* of a game G, denoted by $\mathrm{val}(G)$, is the players' maximum success probability over all possible strategies they may use.

Here is a very natural operation on games, called *parallel repetition*: starting with a k-player game G, we can construct a new k-player game $G^{\otimes n}$, called the n-fold parallel repetition of G. In $G^{\otimes n}$, the referee will select n independent question tuples $(x_i^1, \ldots, x_i^t)$ from μ for each *coordinate* $i = 1, \ldots, n$, and send $(x_1^t, \ldots, x_n^t)$ to each player t. Each player has to respond with n answers, and they win this repeated game if their answers and questions for each coordinate i satisfy the original game predicate V. We call G the *base game* of the parallel repeated game $G^{\otimes n}$.

The central question we consider is how $\mathrm{val}(G^{\otimes n})$ depends on the base game G and the number of repetitions n. When G is a two-player game, the behavior of $\mathrm{val}(G^{\otimes n})$ has been extensively studied, especially due to its applications to probabilistically checkable proofs and hardness of approximation. The central result in this area is Raz's Parallel Repetition Theorem [11] (coupled with subsequent improvements due to Holenstein [8]), which states the following:

▶ **Theorem 1** (Two-player parallel repetition)**.** *Let G be a one-round two-player game with $\mathrm{val}(G) \leq 1 - \varepsilon$ for some $\varepsilon \in (0, 1)$. Then for all $n \geq 0$,*

$$\mathrm{val}(G^{\otimes n}) \leq \exp\left(-\frac{c\varepsilon^3 n}{\log |\mathcal{A}|}\right)$$

where $|\mathcal{A}|$ is the answer alphabet size of G and $c > 0$ is a universal constant.

In other words, for *nontrivial* two-player games G (i.e., games whose value is less than 1), $\mathrm{val}(G^{\otimes n})$ decays exponentially fast in n.

What about parallel repetition for games involving more than two players? It remains an intriguing open question whether Raz's Theorem can be extended to the multiplayer case. An early result of Verbitsky [13] shows that for multiplayer games G with $\mathrm{val}(G) < 1$, the value of the repeated game $G^{\otimes n}$ must decay to 0 as n goes to infinity. However, the bound on the rate of decay is extremely weak: his result only shows that $\mathrm{val}(G^{\otimes n})$ is bounded by a function that is inversely proportional to the inverse Ackermann function of n [10]! This poor rate of decay comes from its black-box usage of the density Hales-Jewett theorem from extremal combinatorics.

So far, Verbitsky's theorem is still the only result that gives a general parallel repetition bound for all multiplayer games. Exponential-decay parallel repetition bounds (à la Raz) for multiplayer games have been proven when there are additional assumptions on the game; for example, it has long been a folklore result that multiplayer *free* games satisfy an exponential-decay parallel repetition theorem [4].[1] Recently, Bavarian, Vidick and Yuen [1] studied a variant of parallel repetition (called "anchoring") where the base game G is first modified to an equivalent game $\widetilde{G}$ before being repeated in parallel, producing $\widetilde{G}^{\otimes n}$. They proved that the value of $\widetilde{G}^{\otimes n}$ is exponentially small in n when $\mathrm{val}(G) < 1$, and otherwise $\mathrm{val}(\widetilde{G}^{\otimes n}) = 1$.[2]

We observe that the class of multiplayer games for which we have exponential-decay parallel repetition bounds (or, for that matter, any rate of decay better than inverse Ackermann!) all share a particular feature in common: when viewed as hypergraphs, the games

[1] A free game is one where each players' question is independent of all the other players'.

[2] In fact, Bavarian, Vidick and Yuen were motivated by the question of parallel repetition for *quantum* players; they showed that so-called "anchored" games satisfy quantum parallel repetition theorems.

all possess a certain "well-connectedness" property. For example, consider any two question tuples $x, \widehat{x}$ in the support of the question distribution μ of a free game. The question tuple $x = (x^1, \ldots, x^k)$ can be "locally morphed" to $\widehat{x} = (\widehat{x}^1, \ldots, \widehat{x}^k)$ via a sequence of question tuples $(\widehat{x}^1, \ldots, \widehat{x}^j, x^{j+1}, \ldots, x^k)$ for $j = 1, \ldots, k$, each of which remain in the support of μ. Furthermore, the anchoring transformations of [1] can be understood as improving the connectivity properties of the base game before repetition. In this paper, we formalize this well-connectedness property as a type of *expansion* of the base game, and show that any connected multiplayer game has exponential-decay parallel repetition bounds. We associate with every base game G, a related graph H_G (see Definition 2) and show that if H_G is connected, then the value of the repeated game, $\mathrm{val}(G^{\otimes n})$, goes down exponentially in n, more precisely, for sufficiently large n,

$$\mathrm{val}(G^{\otimes n}) \leq \exp\left(-\frac{c\varepsilon^5\lambda^2 n}{\log|\mathcal{A}|}\right),$$

where λ is the spectral gap of the Laplacian of the graph H_G and c is some universal constant (see Theorem 5 for an exact statement of the result). Thus, if the graph H_G is connected (i.e., $\lambda > 0$), we have an exponential decay in n. In the case of games G, wherein the associated graph H_G is not only connected but also expanding (i.e., λ is a constant), as is the case with free games, and anchored games, the rate of exponential decay is a function of alphabet size $|\mathcal{A}|$ of the base game G as in Raz's theorem.

Why care about games with more than two players?

The notion of a game is an extremely basic notion, and it's use is pervasive in communication complexity, probabilistically checkable proofs (PCPs), etc. Whereas two-player games are already quite powerful and give us a lot, many problems are inherently higher-dimensional, i.e., would more naturally be cast as games with more than two players. The reason this is not commonly done is because we don't know how to analyze these creatures. For example, constraint-satisfaction-problems with arity k are naturally cast as a k-player game. They can be reduced to a two-player game in the same way that a hypergraph can be converted to a graph, but this reduction in dimensionality might be lossy. Indeed, it is empirically true that PCPs with 3 or more queries are much more powerful than 2-query PCPs, but what is the reason for this?

Furthermore, this sudden jump in difficulty in going from two-player problems to three or more players is encountered also when studying multiparty communication complexity, and seemingly because of the same technique limitations. While direct sum and direct product theorems are known for two-party communication complexity, nothing is known for the multiparty setting (in the so-called *number-on-forehead* model), and in fact making progress on this is connected to hard problems in circuit complexity.

We feel that the study of games with three or more players is a very important component in understanding such questions.

1.1 Notation

We establish some notational conventions, before stating our results formally.

For a k-player game G, we will let $\mathcal{X}^t$ denote the question alphabet for player t, and $\mathcal{X} = \mathcal{X}^1 \times \mathcal{X}^2 \times \cdots \times \mathcal{X}^k$ is the question alphabet for all the players together, underlying the question distribution μ. We will let $\mathcal{A}^t$ denote the answer alphabet for player t, and $\mathcal{A} = \mathcal{A}^1 \times \cdots \times \mathcal{A}^k$ to denote the answer alphabet for all the players together.

We will use superscripts to denote the players, and subscripts to denote the coordinate in parallel repetition. For example, x_i^t denotes the question received by player t in coordinate i. A single variable x can denote questions to the players in some coordinate clear from context, or a single coordinate game. We use x^{-t} to denote the questions to all but the t-th player in a single coordinate. When talking about multiple coordinates, we will use subscripts: x_{-i} denotes the questions to players in all but the i'th coordinate, and x_{-i}^t denotes the all questions to player t in the repeated game except for the i'th coordinate. To denote the question to player t in all coordinates, we use $x_{[n]}^t$.

We largely adopt the notational conventions from [8] for probability distributions. We let capital letters denote random variables and lower case letters denote specific samples. We use P_X to denote the probability distribution of random variable X, and $\mathsf{P}_X(x)$ to denote the probability that $X = x$ for some value x. For multiple random variables, e.g., X, Y, Z, $\mathsf{P}_{XYZ}(x, y, z)$ denotes their joint distribution with respect to some probability space understood from context.

We use $\mathsf{P}_{Y|X=x}(y)$ to denote the conditional distribution $\mathsf{P}_{YX}(y, x)/\mathsf{P}_X(x)$, which is defined when $\mathsf{P}_X(x) > 0$. When conditioning on many variables, we usually use the shorthand $\mathsf{P}_{X|y,z}$ to denote the distribution $\mathsf{P}_{X|Y=y,Z=z}$. For an event W we let $\mathsf{P}_{XY|W}$ denote the distribution conditioned on W. We use the notation $\mathbb{E}_X f(x)$ and $\mathbb{E}_{\mathsf{P}_X} f(x)$ to denote the expectation $\sum_x \mathsf{P}_X(x) f(x)$.

Let P_{X_0} be a distribution over $\mathcal{X}$ and $\mathsf{P}_{X_1,Y}$ a joint distribution over $\mathcal{X} \times \mathcal{Y}$. Suppose for every x in the support of P_{X_0}, the conditional distribution $\mathsf{P}_{Y|X_1=x}$ defined over $\mathcal{Y}$ is well-defined. We then define the distribution $\mathsf{P}_{X_0}\mathsf{P}_{Y|X_1}$ over $\mathcal{X} \times \mathcal{Y}$ as

$$(\mathsf{P}_{X_0}\mathsf{P}_{Y|X_1})(x, y) := \mathsf{P}_{X_0}(x) \cdot \mathsf{P}_{Y|X_1=x}(y).$$

For two random variables X_0 and X_1 over the same set $\mathcal{X}$, we use

$$\|\mathsf{P}_{X_0} - \mathsf{P}_{X_1}\| := \frac{1}{2} \sum_{x \in \mathcal{X}} |\mathsf{P}_{X_0}(x) - \mathsf{P}_{X_1}(x)|,$$

to denote the total variation distance between P_{X_0} and P_{X_1}.

1.2 Our results

To define our class of connected and expanding games, we need the following notion of the $(k-1)$-*connection graph* of a game G. This graph, denoted H_G, has a vertex for every k-tuple of questions, and two k-tuples are connected by an edge if they agree on $(k-1)$ coordinates. A game is $(k-1)$-connected iff H_G is connected.

To further define our notion of expansion for a k-player game we need to take the weights of G into account when defining H_G. For this it is instructive to think of an intermediate bipartite graph $B_G = (\mathcal{X}', \mathcal{X}, E)$ as follows. The right hand vertices is simply $\mathcal{X}$, the set of all k-tuples of questions, and we endow these vertices with weights as given by G. The left hand vertices consists of all punctured k-tuples, which are k-tuples of questions where exactly one of the entries is replaced by a special $\star$ symbol. Connect each k-tuple of questions to all of the k ways to make it into a punctured k-tuple. Now, consider the distribution on punctured tuples obtained by selecting a random k-tuple from $\mathcal{X}$ according to the game distribution, and then puncturing it in a random location. The graph H_G is defined by selecting a random punctured tuple according to this distribution, and then selecting independently two k-tuples conditioned on this puncturing. Note that each completion is distributed exactly according to the original game distribution.

We now move to a completely explicit description consistent with the above. In what follows, $\mathsf{P}_X(x)$ denotes the probability of question tuple x under the question distribution μ, $\mathsf{P}_{X^t}(x^t)$ denotes the marginal probability of player t's question, and $P_{X^t|X^{-t}=x^{-t}}(x^t)$ denotes the same probability, conditioned on the other players having received x^{-t}.

▶ **Definition 2** ($(k-1)$-connection graph of G)**.** Let $G = (\mu, V)$ be a k-player game with question set $\mathcal{X} = \mathcal{X}^1 \times \cdots \mathcal{X}^k$. The **$(k-1)$-connection graph of G** is the weighted graph $H_G = (V_H, \rho)$ with vertex set $V_H = \mathcal{X}$ and weight function $\rho : \mathcal{X} \times \mathcal{X} \to [0,1]$, defined as follows: for every $x, x' \in \mathcal{X}$,

$$\rho(x,x') = \begin{cases} \frac{1}{k}\,\mathsf{P}_X(x)\left[\sum_{t\in[k]}\mathsf{P}_{X^t|x^{-t}}(x^t)\right] & \text{if } x = x', \\ \frac{1}{k}\,\mathsf{P}_{X^{-t}}(x^{-t})\cdot\mathsf{P}_{X^t|x^{-t}}(x^t)\cdot\mathsf{P}_{X^t|x^{-t}}(x'^t) & \text{if } \exists t \text{ s.t. } x^{-t} = x'^{-t},\ x^t \neq x'^t, \\ 0 & \text{otherwise.} \end{cases}$$

The weight function $\rho(x,x')$ can be viewed as the probability of generating the pair (x,x') according to the following random process: first, $x \in \mathcal{X}$ is sampled from the distribution P_X. Then, a coordinate $t \in [k]$ is chosen uniformly at random, and x' is sampled from the conditional distribution $\mathsf{P}_{X|x^{-t}}$ (that is, the distribution μ conditioned on x^{-t}).

Observe that ρ is symmetric, i.e., $\rho(x,x') = \rho(x',x)$. Furthermore, note that the weight on any given vertex is exactly:

$$\begin{aligned}\rho(x,\cdot) &= \sum_{x'}\rho(x,x') = \\ &= \mathsf{P}_X(x)\cdot\frac{1}{k}\sum_{t\in[k]}\mathsf{P}_{X^t|x^{-t}}(x^t) + \mathsf{P}_X(x)\cdot\frac{1}{k}\sum_{t\in[k]}\sum_{x'^t\neq x^t}\mathsf{P}_{X^t|x^{-t}}(x'^t) \\ &= \mathsf{P}_X(x).\end{aligned}$$

Therefore $\rho(\cdot,\cdot)$ is a probability distribution over $\mathcal{X}\times\mathcal{X}$.

▶ Remark. Henceforth, when we talk about graph properties such as diameter, connectedness or expansion of H_G, we will do so only with respect to the vertices having non-zero weight.

We now recall the definition of a graph with a weight function ρ being a spectral expander:

▶ **Definition 3** (Normalized Laplacian)**.** Let H be a weighted graph where $\rho(u,v) \le 1$ is the weight between vertices u and v. The normalized Laplacian $L_H \in \mathbb{R}^{|V|\times|V|}$ of H is defined to be

$$(L_H)_{u,v} = \begin{cases} 1 - \frac{\rho(u,v)}{\rho(v)} & \text{if } u = v \text{ and } \rho(v) \neq 0 \\ -\frac{\rho(u,v)}{\sqrt{\rho(u)\rho(v)}} & \text{if } \rho(u),\rho(v) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\rho(u) = \sum_v \rho(u,v)$ and $\rho(v) = \sum_u \rho(u,v)$.

It is well-known that the second smallest eigenvalue of H is given by the following variational formula: for all $r \in \mathbb{N}$,

$$\lambda(H) = \inf_g \frac{\sum_{u,v}\rho(u,v)\|g(u)-g(v)\|^2}{\sum_u \rho(u)\|g(u)-\overline{g}\|^2} \tag{1}$$

where the infimum is over all vector-valued functions $g : V(H) \to \mathbb{R}^r$ defined on the vertices of H, and $\overline{g}$ is a vector in $\mathbb{R}^r$ where for each $i \in [r]$, $\overline{g}_i = \sum_u \rho(u) g(u)_i$.

▶ **Definition 4** (Expander graph). Let $\lambda \in (0,1)$. A graph H is a λ-expander if $\lambda(H) \geq \lambda$.

Our main result is an exponential-decay parallel repetition bound for multiplayer games whose $(k-1)$-connection graph is expanding:

▶ **Theorem 5** (Main theorem). *Let $\epsilon, \lambda \in (0,1)$. Let G be a k-player game with $\mathrm{val}(G) \leq 1-\varepsilon$. If the $(k-1)$-connection graph H_G is a λ-expander, then we have, for all $n \geq \frac{\log 4/\varepsilon}{\varepsilon^5 \lambda^2}$:*

$$\mathrm{val}(G^{\otimes n}) \leq \exp\left(-\frac{c\varepsilon^5\lambda^2 n}{\log|\mathcal{A}|}\right)$$

where $\mathcal{A} = \mathcal{A}^1 \times \cdots \times \mathcal{A}^k$ is the answer alphabet in G, and c is a universal constant.

By applying our main theorem to free games and anchored games, we recover existing exponential-decay parallel repetition results for multiplayer games [4, 1]. We also get an exponential-decay lower bound for *connected games* – games whose $(k-1)$-connection graph is connected. We record these consequences in the following corollary:

▶ **Corollary 6.** *Let G be a k-player game with $\mathrm{val}(G) \leq 1-\varepsilon$, and let $n \geq \frac{\log 4/\varepsilon}{\varepsilon^5 \lambda^2}$. If G is:*

1. *Free, i.e., $\mu(x) = \mu^1(x^1) \times \cdots \times \mu^k(x^k)$, then*

$$\mathrm{val}(G^{\otimes n}) \leq \exp\left(-\frac{c\,\varepsilon^5\,n}{k^2 \log|\mathcal{A}|}\right).$$

2. *α-Anchored (see Definition 12, and [1] for more details), then*

$$\mathrm{val}(G^{\otimes n}) \leq \exp\left(-\frac{c\,\alpha^{2k}\,\varepsilon^5\,n}{64\ k^2 \log|\mathcal{A}|}\right).$$

3. *Connected, i.e., the $(k-1)$-connection graph is connected, then*

$$\mathrm{val}(G^{\otimes n}) \leq \exp\left(-\frac{c\,\rho_{min}^2\,\varepsilon^5\,n}{\log|\mathcal{A}|}\right)$$

   where $\rho_{min} = \min_{u,v:\rho(u,v)>0} \rho(u,v)$. In particular, if the game G is such that μ is the uniform distribution over some set $S \subseteq \mathcal{X}$, then $\rho_{min} \geq (k|S|^2)^{-1}$.

where c is a universal constant.

The proof of Corollary 6 can be found in Appendix A.

Observe that our proof of exponential decay for games whose corresponding $(k-1)$-connection graph is connected proves a rate of exponential decay that is dependent on the size of the the base game G. It is conceivable that this rate of decay can be further improved to depend only on the alphabet size $|\mathcal{A}|$ of the base game and be independent of the size of the base game (as is the case in Raz's theorem for 2 player games). For games whose corresponding $(k-1)$-connection graph is expanding (as is the case with free games and anchoring games), we obtain a rate of exponential decay which is a function of only the base game's alphabet size.

▶ Remark. For simplicity, we state Theorem 5 assuming the base game has a connected $(k-1)$-connection graph. It is easy to check (from the proof of Theorem 5) that it also extends to games that are disjoint union of games each of which has a connected $(k-1)$-connection graph. By disjoint union we mean that each question occurs only in one of the components. For $k = 2$, this captures all possible games since every game is a union of

disjoint games whose ($k-1=2-1=1$)-connection graphs are connected). We note that for 2-player games, there are alternate proof techniques [11, 8] using correlated sampling which prove even better rate of exponential decay (our proof of Theorem 5 does not use correlated sampling). In contrast, for $k > 2$, there are many games that are not captured by our theorem. We will see below an example of such a $k = 3$-player game called the GHZ game.

A comment about fortified games

Bavarian, Vidick and Yuen also proved a parallel repetition bound for a special class of multiplayer games *fortified games* [2] (a class of games introduced by Moshkovitz [9]). However, we do not consider this a "true" exponential-decay parallel repetition bound, because it does not establish a decay bound of the form $\mathrm{val}(G^{\otimes n}) \leq \exp(-\beta n)$ for some constant β that depends on the game G, but is independent of n. Instead, it proves a decay bound that is exponential only for a small number of repetitions (depending on the base game). After this small number of repetitions, there are no guarantees about any further value decay (other than that promised by Verbitsky's theorem). Because we are interested in the asymptotic behavior of an n-repeated multiplayer game as n goes to infinity, we do not consider the parallel repetition of fortified games here.

A disconnected three-player game

It may seem that, given Corollary 6, we have established a general exponential-decay parallel repetition bound for *all* multiplayer games, albeit with some slightly annoying dependency on a quantity related to the minimum probability of any question from μ. Unfortunately, this is far from the case.

Here is a simple three-player game called the *GHZ game* whose parallel repetition resists analysis; the best decay bound we have comes from Verbitsky's theorem [13]. The GHZ game is a three-player game[3] where the referee samples a question triple (x, y, z) uniformly at random from $\{(1,0,0),(0,1,0),(0,0,1),(1,1,1)\}$, and sends each bit of the triple to the corresponding player. The players respond with bits a, b, c respectively, and they win iff $x \wedge y \wedge z = a \oplus b \oplus c$. It is easy to see that $\mathrm{val}(GHZ) = 3/4$ (achieved by the strategy where all players always output "0"). However, the best general bound we have on $\mathrm{val}(GHZ^{\otimes n})$ is the weak inverse-Ackermann decay given by Verbitsky's theorem.

Our main theorem does not apply because the $(k-1)$-connection graph H_{GHZ} of the GHZ game is actually *disconnected*; no two question triples are connected via a single coordinate change. One necessary criterion for the $(k-1)$-connection graph to be connected is that, after fixing any subset of $(k-1)$ players' questions, the remaining player's question is yet undetermined. On the other hand, the players' questions in the GHZ game satisfy a linear relation (i.e. $x \oplus y \oplus z = 1$), and thus fixing two players' questions also fixes the third.

We believe that the strong correlations present in the GHZ question distribution represent the "hardest instance" of the multiplayer parallel repetition problem. Existing techniques from the two-player case (which we leverage in this paper) appear to be incapable of analyzing games with question distributions with such strong correlations. Thus we explicitly

[3] The GHZ game comes from the study of non-locality in quantum physics; when the players use classical strategies, their maximum success probability is $\mathrm{val}(G) = 3/4$, but using quantum entanglement, the GHZ can be won with certainty [6].

raise the open question of proving an exponential-decay parallel repetition bound for the GHZ game:

▶ **Conjecture 7** (GHZ parallel repetition)**.** There exists a constant $\beta > 0$ such that for all n,

$$\mathrm{val}(GHZ^{\otimes n}) \leq \exp(-\beta n).$$

Finally, we remark that this challenge of handling strongly correlated question distributions is reminiscent of the challenge of proving *direct sum* theorems for multiparty communication complexity in the *Number-on-Forehead* (NOF) model. There, each player sees every players' inputs but their own, so fixing $(k-1)$ out of k players' inputs will fix the remaining player's inputs. Proving direct sum results in NOF communication complexity has resisted progress for reasons that appear to be related to the multiplayer parallel repetition problem.

2 Proof of Theorem 5

2.1 Proof outline

We first give a brief overview of the information-theoretic approach to proving two-player parallel repetition as in [11, 8], and explain the technical barrier to extending the proof to three or players. We then will describe how we circumvent this technical barrier.

Essentially all known proofs of parallel repetition proceed via reduction, showing how a "too good" strategy for the repeated game G^n can be "rounded" into a strategy for G with success probability strictly greater than $\mathrm{val}(G)$, yielding a contradiction.

Let $\mathcal{S}^n$ be a strategy for G^n that has a high success probability. Either by induction or via a probabilistic argument one can identify a set of coordinates S and an index i such that $\Pr(\text{Players win round } i | W_S) > \mathrm{val}(G) + \delta$, where W is the event that the players' answers satisfy the predicate V in all instances of G indexed by S. Given a pair of questions (x, y) in G the strategy $\mathcal{S}$ embeds them in the i-th coordinate of a n-tuple of questions

$$x_{[n]}y_{[n]} = \begin{pmatrix} x_1, x_2, \ldots, x_{i-1}, & x & , x_{i+1}, \ldots, x_n \\ y_1, y_2, \ldots, y_{i-1}, & y & , y_{i+1}, \ldots, y_n \end{pmatrix}$$

that is distributed according to $\mathsf{P}_{X_{[n]}Y_{[n]}|X_i=x,Y_i=y,W}$. The players then simulate $\mathcal{S}^n$ on $x_{[n]}$ and $y_{[n]}$ respectively to obtain answers $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$, and return (a_i, b_i) as their answers in G. The single-shot strategy $\mathcal{S}$ succeeds with probability precisely $\Pr(\text{Win } i | W_S)$ in G, yielding the desired contradiction.

As $\mathcal{S}^n$ need not be a product strategy, conditioning on W_S may introduce correlations that make $\mathsf{P}_{X_{[n]}Y_{[n]}|X_i=x,Y_i=y,W_S}$ impossible to sample exactly. A key insight in Raz' proof of parallel repetition is that it is still possible for the players to *approximately* sample from this distribution. For this, we introduce a *dependency-breaking variable* R with the following properties:

(a) Given $r \sim \mathsf{P}_R$ the players can locally sample $x_{[n]}$ and $y_{[n]}$ according to $\mathsf{P}_{X_{[n]}Y_{[n]}|X_i=x,Y_i=y,W_S}$,

(b) The players can jointly sample from P_R using shared randomness.

In [8] R is defined so that a sample r fixes at least one of $\{x_{i'}, y_{i'}\}$ for each $i' \neq i$. It can then be shown that conditioned on x, R is nearly (though not exactly) independent of y, and vice-versa. In other words,

$$\mathsf{P}_{R|X_i=x,W_S} \approx \mathsf{P}_{R|X_i=x,Y_i=y,W_S} \approx \mathsf{P}_{R|Y_i=y,W_S} \tag{2}$$

where "≈" denotes closeness in statistical distance. Eq. (2) suffices to guarantee that the players can *approximately* sample the same r from $\mathsf{P}_{R|X_i=x,Y_i=y,W_S}$ with high probability, achieving point (b) above. This sampling is accomplished through a technique called *correlated sampling*.

This argument relies heavily on the assumption that there are only two players who employ a deterministic strategy. With more than two players, it is not known how to design an appropriate dependency-breaking variable R that satisfies *both* items (a) and (b) above: in order to be jointly sampleable, R needs to fix as few inputs as possible; in particular, no single player should require knowledge of the other player's questions to sample R. On the other hand, in order to allow players to locally sample their inputs conditioned on R, the variable needs to fix as many inputs as possible. These two requirements turn out to be in direct conflict as soon as there are more than two players, and a straightforward generalization of the two-player version of the dependency-breaking variable cannot be "correlatedly sampled" by all players, unless every player has knowledge of the question received by some other player.

We avoid this roadblock by proving in Section 2.4 that if the $(k-1)$-connected graph H_G is connected, then the players can avoid correlated sampling altogether. In fact, they can sample an appropriate dependency-breaking variable from a *global* distribution that does not depend on any player's question.

2.2 Following Raz-Holenstein

Fix a k-player game $G = (\mu, V)$, with answer alphabet $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_k$ and $\mathrm{val}(G) = 1-\varepsilon$. Consider the n-fold parallel repetition $G^{\otimes n}$ and consider an optimal strategy $\{f^t : (\mathcal{X}^t)^{\otimes n} \to (\mathcal{A}^t)^{\otimes n}\}_{t\in[k]}$ for the k players.

For $i \in [n]$, let W_i denote the event that the players win coordinate i using this optimal strategy. Let $W = W_1 \wedge \cdots \wedge W_n$ denote the event that the players win all coordinates. For a set $S \subseteq [n]$, let $W_S = \wedge_{i\in S} W_i$. In the following, all probabilities are with respect to this optimal strategy.

▶ **Proposition 8.** *Let $\varepsilon > 0$. Suppose that $\log 1/\Pr(W) \le \varepsilon n/16 - \log 4/\varepsilon$. Then there exists a set $S \subseteq [n]$ of size at most $t = \frac{8}{\varepsilon}(\log 4/\varepsilon + \log 1/\mathsf{P}(W))$ such that*

$$\mathsf{P}_{i\notin S}(\neg W_i | W_S) \le \varepsilon/2$$

where i is chosen uniformly from $[n] - S$.

Proof. Set $\delta = \varepsilon/8$. Let $W_{>1-\delta}$ denote the event that the players won more than $(1-\delta)n$ rounds. To show existence of such a set S, we will show that $\mathbb{E}_S \mathsf{P}(\neg W_i | W_S) \le \varepsilon/2$, where S is a (multi)set of t independently chosen indices in $[n]$. This implies that there exists a particular set S such that $\mathsf{P}(\neg W_i | W_S) \le \varepsilon/2$, which concludes the claim.

First we write, for a fixed S,

$$\begin{aligned}\mathsf{P}(\neg W_i | W_S) = \Pr(\neg W_i | W_S, W_{>1-\delta})\mathsf{P}(W_{>1-\delta}|W_S) \\ + \mathsf{P}(\neg W_i | W_S, \neg W_{>1-\delta})\mathsf{P}(\neg W_{>1-\delta}|W_S).\end{aligned}$$

Observe that $\mathsf{P}(\neg W_i | W_S \wedge W_{>1-\delta})$ is the probability that, conditioned on winning all rounds in S, the randomly selected coordinate $i \in [n] - S$ happens to be one of the (at most) δn lost rounds. This is at most $\delta n/(n-t) \le \varepsilon/4$. Now observe that

$$\mathop{\mathbb{E}}_S \mathsf{P}(\neg W_{>1-\delta} | W_S) \le \mathop{\mathbb{E}}_S \frac{\mathsf{P}(W_S | \neg W_{>1-\delta})}{\mathsf{P}(W_S)} \le \frac{1}{\mathsf{P}(W)}(1-\delta)^t \le \varepsilon/4$$

where for the second inequality we used the fact that $\mathsf{P}(W_S) \ge \mathsf{P}(W)$. ◀

For the remainder of this proof we will fix a set S as given by Proposition 8. By renaming coordinates, we will assume without loss of generality that S is the last t coordinates of $[n]$. We will let $m = n - |S|$. We will refer to the games indexed by set S as the S-games.

2.3 Dependency-breaking variables

We define the k-player analogue of the dependency-breaking variable R that is used so crucially in information-theoretic proofs of parallel repetition [11, 8, 3]. R will consist of a variable Ω, which fixes the questions for the S-games, and at least $(k-1)$-of-k questions in every other coordinate, and a variable $Z = (A_S)$, which fixes the answers of S-games. More formally, $\Omega = (\Omega_1, \ldots, \Omega_m, X_S)$, where X_S are fixed questions for the S-games. Each $\Omega_i = (D_i, M_i)$, for $i \in \overline{S}$, where D_i is a uniformly random value in $[k]$, and

$$M_i = X_i^{-t} \quad \text{if } D_i = t$$

In other words, D_i specifies which player's question to omit; the other $(k-1)$ players are fixed.

For $i \notin S$, we let Ω_{-i} denote Ω with Ω_i omitted. We let $R_{-i} := (\Omega_{-i}, A_S)$. R_i will refer to Ω_i. We will use lowercase letters to denote instantiations of these random variables: e.g., r_{-i}, x_i^t refer to specific values of R_{-i}, X_i^t respectively.

▶ **Claim 9.** *Conditioned on R, $\{X_{[n]}^t\}_{t\in[k]}$ are independent.*

In the following, P_I denotes the distribution of a uniformly random $i \in [m]$, and "$\mathsf{P} \approx_\delta \mathsf{Q}$" indicates that the probability distributions P and Q are δ-close in statistical distance. We will fix

$$\delta = \frac{1}{m}\left(\log \frac{1}{\mathsf{P}(W_S)} + |S| \log |\mathcal{A}|\right).$$

The next lemma states that for an average i, if we sample questions $x_i, \widehat{x}_i$ from the joint probability distribution $\rho(x_i, \widehat{x}_i)$, the distributions of the corresponding dependency-breaking variables will be close.

▶ **Lemma 10.**

$$\frac{1}{m}\sum_i \sum_{x_i, \widehat{x_i} \in \mathcal{X}} \rho(x_i, \widehat{x}_i) \left\| \mathsf{P}_{R_{-i}|x_i, W_S} - \mathsf{P}_{R_{-i}|\widehat{x}_i, W_S} \right\|_1 \leq O(\sqrt{\delta})$$

where $\rho(\cdot,\cdot)$ is the weight function of the $(k-1)$-connection graph H_G.

Proof. First, we establish the following: for all $t \in [k]$, we have

$$\mathop{\mathbb{E}}_i \sum_{x_i^{-t}, x_i^t, \widehat{x}_i^t} \mathsf{P}_{X_i^{-t}}(x_i^{-t})\, \mathsf{P}_{X_i^t|x_i^{-t}}(x_i^t) \cdot \mathsf{P}_{\widehat{X}_i^t|x_i^{-t}}(\widehat{x}_i^t) \left\| \mathsf{P}_{R_{-i}|x_i, W_S} - \mathsf{P}_{R_{-i}|\widehat{x}_i, W_S} \right\|_1 \leq O(\sqrt{\delta}) \tag{3}$$

where we use the shorthand $x_i := x_i^{-t} x_i^t$ and $\widehat{x}_i := x_i^{-t} \widehat{x}_i^t$. This follows from the same arguments found in [8, 3]; for each player t, we can treat the other $(k-1)$ players as one "meta player", and apply the two-player analysis to obtain (3).

Observe that when $x_i^t \neq \widehat{x}_i^t$, we have

$$\mathsf{P}_{X_i^{-t}}(x_i^{-t}) \cdot \mathsf{P}_{X_i^t|x_i^{-t}}(x_i^t) \cdot \mathsf{P}_{\widehat{X}_i^t|x_i^{-t}}(\widehat{x}_i^t) = k\rho(x_i, \widehat{x}_i).$$

On the other hand, when $x_i^t = \widehat{x}_i^t$, $x_i = \widehat{x}_i$ so therefore $\left\|\mathsf{P}_{R_{-i}|x_i,W_S} - \mathsf{P}_{R_{-i}|\widehat{x}_i,W_S}\right\|_1 = 0$. Furthermore, for x_i and $\widehat{x}_i$ that differ in more than 1 coordinate, we have $\rho(x_i,\widehat{x}_i) = 0$, and for every $x_i, \widehat{x}_i$ such that $\rho(x_i,\widehat{x}_i) \neq 0$, there exists a unique $t \in [k]$ such that $x_i^t \neq \widehat{x}_i^t$. Thus we can bound for every i:

$$\sum_{x_i,\widehat{x}_i \in \mathcal{X}} \rho(x_i,\widehat{x}_i) \left\|\mathsf{P}_{R_{-i}|x_i,W_S} - \mathsf{P}_{R_{-i}|\widehat{x}_i,W_S}\right\|_1$$
$$= \frac{1}{k} \sum_{t\in[k]} \sum_{x_i^{-t},x_i^t,\widehat{x}_i^t} \mathsf{P}_{X_i^{-t}}(x_i^{-t}) \cdot \mathsf{P}_{X_i^t|x_i^{-t}}(x_i^t) \cdot \mathsf{P}_{\widehat{X}_i^t|x_i^{-t}}(\widehat{x}_i^t) \left\|\mathsf{P}_{R_{-i}|x_i,W_S} - \mathsf{P}_{R_{-i}|\widehat{x}_i,W_S}\right\|_1 .$$

Averaging over i and using (3), we obtain the statement of the lemma. ◀

2.4 Avoiding correlated sampling using expansion

At this point, ideally, every player would like to sample from $R_{-i}|x_i, W_S$. Lemma 10 establishes that $R_{-i}|x_i, W_S$ is close to $R_{-i}|x_i^{-t}, W_S$ for each $t \in [k]$. None of the players alone has knowledge of x_i^{-t}, however. We will show now that nevertheless, there is a *global* distribution known to all the players, from which the players can approximately sample $R_{-i}|x_i, W_S$.

▶ **Lemma 11.** *For all $i \in [m]$ there exists a distribution $\widetilde{\mathsf{P}}_{R_{-i}}$ over R_{-i} such that*

$$\frac{1}{m}\sum_i \sum_x \rho(x) \|\mathsf{P}_{R_{-i}|x,W_S} - \widetilde{\mathsf{P}}_{R_{-i}}\|_1 \leq O\left(\frac{\delta^{1/4}}{\sqrt{\lambda}}\right).$$

Proof. For each i, define the vector-valued function $g_i : \mathcal{X} \to \mathbb{R}^{R_{-i}}$ as follows: for all $x \in \mathcal{X}$,[4]

$$g_i(x) = \sqrt{\mathsf{P}_{R_{-i}|x,W_S}}$$

where $\sqrt{\mathsf{P}_{R_{-i}|x,W_S}}$ denotes the entry-wise square root of the probability distribution $\mathsf{P}_{R_{-i}|x,W_S}$, viewed as a vector. In other words, the entries of $g_i(x)$ are indexed by different values r_{-i} of the random variable R_{-i}. Thus, g_i is a unit vector in the ℓ_2 norm.

For any i and any $x, \widehat{x} \in \mathcal{X}$, the quantity $\|g_i(x) - g_i(\widehat{x})\|^2$ is simply the square of the *Hellinger distance* between $\mathsf{P}_{R_{-i}|x,W_S}$ and $\mathsf{P}_{R_{-i}|\widehat{x},W_S}$, which can be related to their statistical distance by

$$\|g_i(x) - g_i(\widehat{x})\|^2 \leq \|\mathsf{P}_{R_{-i}|x,W_S} - \mathsf{P}_{R_{-i}|\widehat{x},W_S}\|_1.$$

By Lemma 10, we can average the above inequality over all i and choosing $x, \widehat{x}$ according to the probability distribution $\rho(x,x')$, we get

$$\frac{1}{m}\sum_i \sum_{x,\widehat{x}} \rho(x,\widehat{x}) \|g_i(x) - g_i(\widehat{x})\|^2 \leq \mathop{\mathbb{E}}_i \sum_{x,\widehat{x}} \rho(x,\widehat{x}) \|\mathsf{P}_{R_{-i}|x,W_S} - \mathsf{P}_{R_{-i}|\widehat{x},W_S}\|_1 \leq O(\sqrt{\delta})$$

But now we can leverage Equation (1). For every i, define the vector $\overline{g}_i = \sum_x \mathsf{P}_X(x) g_i(x)$. This is not necessarily a unit vector, but we have the relation

$$\frac{1}{m}\sum_i \sum_x \rho(x) \|g_i(x) - \overline{g}_i\|^2 \leq \frac{1}{\lambda m} \sum_i \sum_{x,\widehat{x}} \rho(x,\widehat{x}) \ \|g_i(x) - g_i(\widehat{x})\|^2 \leq O\left(\frac{\sqrt{\delta}}{\lambda}\right).$$

[4] Here, when we write x, we are implicitly mean x_i; we drop the subscript i for notational convenience.

If $O(\sqrt{\delta}/\lambda)$ is small, then this implies that on average, the vectors $g_i(x)$ are all close to a fixed state $\overline{g}_i$. Since $g_i(x)$ are all unit vectors, this implies that $\overline{g}_i$ is close to a unit vector. By increasing the error by a constant factor, we can assume that $\overline{g}_i$ is in fact a unit vector. Thus we can construct the probability distribution

$$\widetilde{\mathsf{P}}_{R_{-i}}(r_{-i}) = \overline{g}_i(r_{-i})^2.$$

Using that the statistical distance is at most (up to constant factors) the square root of the Hellinger distance, we get that

$$\frac{1}{m}\sum_i \sum_x \rho(x) \|\mathsf{P}_{R_{-i}|x,W_S} - \widetilde{\mathsf{P}}_{R_{-i}}\|_1 \leq O(\delta^{1/4}\lambda^{-1/2}). \qquad \blacktriangleleft$$

2.5 Finishing the proof

Let $\{f^t\}$ be an optimal strategy for the game $G^{\otimes n}$. If $\mathsf{P}(W) \leq \frac{4}{\varepsilon}2^{-\varepsilon n/16}$, then we are done. Otherwise, suppose $\log 1/\mathsf{P}(W) \leq \varepsilon n/16 - \log 4/\varepsilon$. Let the subset S be as given by Proposition 8, and assume the coordinates are numbered so that S is the last $|S|$ coordinates of $[n]$. For all $i \in [m]$, let $\widetilde{\mathsf{P}}_{R_{-i}}$ be as given by Lemma 11. Consider the following single-shot strategy by the players, where x is drawn from μ and x^t is given to player t:

1. Using shared randomness, the players sample an $i \in [m]$ uniformly at random, and sample r_{-i} from $\widetilde{\mathsf{P}}_{R_{-i}}$. Each player t then sets x_i^t to be their "true" question x^t they received from the referee.
2. Using private randomness, each player t samples x_{-i}^t from $\mathsf{P}_{X^t_{-i}|x_i^t,r_{-i}}$. That is, each player samples questions for the n coordinates that come from the repeated game, conditioned on their own true input x_i^t and the dependency-breaking variable r_{-i}.
3. Player t outputs the i'th component of the answer vector $f^t(x^t_{[n]})$.

Lemma 11 implies that after the first step, the sample r_{-i} each player possesses will be, up to statistical error $O(\delta^{1/4}/\sqrt{\lambda})$, distributed according to $\mathsf{P}_{R_{-i}|x,W_S}$ (on average over i and x). Then, by Claim 9, the joint distribution of the random variables $\{X^t_{[n]}\}$ that the players have sampled is

$$\mathsf{P}_{X^1_{[n]}|x_i^t,r_{-i}} \times \cdots \times \mathsf{P}_{X^k_{[n]}|x_i^t,r_{-i}} = \mathsf{P}_{X_{[n]}|x_i,r_{-i}}.$$

Thus, conditioned on r_{-i} and x_i, the distribution of their answers a_i will be distributed according to $\mathsf{P}_{A_i|x_i,r_{-i}}$. Averaging over i, x_i, and r_{-i}, we get that their answers are $O(\delta^{1/4}/\sqrt{\lambda})$-close to being distributed according to

$$\mathsf{P}_I \cdot \mathsf{P}_{X_i} \cdot \mathsf{P}_{R_{-i}|X_i,W_S} \cdot \mathsf{P}_{A_i|X_i,R_{-i}}$$

where P_I stands for the uniform distribution over $i \in [m]$. We also have that, on average i, $\mathsf{P}_{X_i|W_S}$ is $O(\sqrt{\delta})$-close in statistical distance to P_{X_i}. Thus their answers are $O(\delta^{1/4}/\sqrt{\lambda}) + O(\sqrt{\delta})$ close to being distributed as

$$\mathsf{P}_I \cdot \mathsf{P}_{A_i|W_S}.$$

Thus by Proposition 8, the probability that the players win G is at least

$$1 - \varepsilon/2 - \left(O(\delta^{1/4}/\sqrt{\lambda}) + O(\sqrt{\delta})\right).$$

If $O(\delta^{1/4}/\sqrt{\lambda}) + O(\sqrt{\delta}) < \varepsilon/2$, then we would contradict the fact that $\mathrm{val}(G) = 1 - \varepsilon$. This implies that we must have $\delta = \Omega(\varepsilon^4\lambda^2)$. If we let $\mathsf{P}(W) = 2^{-\gamma n}$, then we can write

$$\delta \leq \frac{16}{\varepsilon}\left[\frac{1}{n}\log\frac{4}{\varepsilon} + 2\log|\mathcal{A}|\gamma\right]$$

where we plugged in the bound on $|S| \leq n/2$ from Proposition 8. This implies the lower bound

$$\gamma \geq \Omega\left(\frac{\varepsilon^5\lambda^2}{\log|\mathcal{A}|}\right) \tag{4}$$

when $n \geq \frac{\log 4/\varepsilon}{\varepsilon^5\lambda^2}$, proving the theorem.

References

1. Mohammad Bavarian, Thomas Vidick, and Henry Yuen. Anchoring games for parallel repetition. (manuscript), 2015. arXiv:1509.07466.
2. Mohammad Bavarian, Thomas Vidick, and Henry Yuen. Parallel repetition via fortification: analytic view and the quantum case. (manuscript), 2016. arXiv:1603.05349.
3. Mark Braverman and Ankit Garg. Small value parallel repetition for general games. In *Proc. 47th ACM Symp. on Theory of Computing (STOC)*, pages 335–340, 2015. doi:10.1145/2746539.2746565.
4. Kai-Min Chung, Xiaodi Wu, and Henry Yuen. Parallel repetition for entangled k-player games via fast quantum search. In *Proc. 30th Comput. Complexity Conf.*, pages 512–536, 2015. arXiv:1501.00033, doi:10.4230/LIPIcs.CCC.2015.512.
5. Persi Diaconis and Daniel Stroock. Geometric bounds for eigenvalues of Markov chains. *Ann. Appl. Probab.*, 1(1):36–61, 1991. doi:10.1214/aoap/1177005980.
6. Daniel M. Greenberger, Michael A. Horne, Abner Shimony, and Anton Zeilinger. Bell's theorem without inequalities. *Am. J. Phys.*, 58(12):1131–1143, 1990. doi:10.1119/1.16243.
7. Venkatesan Guruswami. Rapidly mixing Markov chains: a comparison of techniques. (survey), 2016. arXiv:1603.01512.
8. Thomas Holenstein. Parallel repetition: Simplification and the no-signaling case. *Theory Comput.*, 5(1):141–172, 2009. (Preliminary version in *39th STOC*, 2007). arXiv:cs/0607139, doi:10.4086/toc.2009.v005a008.
9. Dana Moshkovitz. Parallel repetition from fortification. In *Proc. 55th IEEE Symp. on Foundations of Comp. Science (FOCS)*, pages 414–423, 2014. doi:10.1109/FOCS.2014.51.
10. D. H. J. Polymath. A new proof of the density Hales-Jewett theorem. *Ann. of Math.*, 175:1283–1327, 2012. arXiv:0910.3926, doi:10.4007/annals.2012.175.3.6.
11. Ran Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, June 1998. (Preliminary version in *27th STOC*, 1995). doi:10.1137/S0097539795280895.
12. Alistair Sinclair. Improved bounds for mixing rates of Markov chains and multicommodity flow. *Combin. Probab. Comput.*, 1(4):351–370, 1992. (Preliminary version in *1st LATIN*, 1992). doi:10.1017/S0963548300000390.
13. Oleg Verbitsky. Towards the parallel repetition conjecture. *Theoret. Comput. Sci.*, 157(2):277–282, 1996. (Preliminary version in *9th Structure in Complexity Theory Conference*, 1994). doi:10.1016/0304-3975(95)00165-4.

A Proof of Corollary 6

For each type of game, we compute a lower bound on the second-smallest eigenvalue of the corresponding $(k-1)$-connection graph. Applying Theorem 5 then yields the statements of the corollary.

A.1 Free games

For simplicity, assume that $\mu(x)$ is the uniform distribution over $[d]^k$, where $d = |\mathcal{X}^1| = \cdots = |\mathcal{X}^k|$.[5] Then the $(k-1)$-connection graph is a weighted version of the d-ary, k-dimensional hypercube (with self loops). Indeed, the corresponding weight function ρ behaves as follows: for $x, x' \in [d]^k$, we have $\rho(x,x) = d^{-(k+1)}$, and $\rho(x,x') = d^{-(k+1)}/k$ when x and x' differ in exactly one coordinate, and is 0 otherwise. If we compute the normalized Laplacian L_H, we get that

$$(L_H)_{u,v} = \begin{cases} 1 - \frac{1}{d} & \text{if } u = v \text{ and } \rho(v) \neq 0 \\ -\frac{1}{kd} & \text{if } \rho(u), \rho(v) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

This is the normalized Laplacian corresponding to the Cayley graph over the Abelian group $(\mathbb{Z}/d\mathbb{Z})^k$ with (weighted) generators $\{g \in (\mathbb{Z}/d\mathbb{Z})^k : |g| \leq 1\}$ where $|g|$ is the number of non-zero components of g. If $g = (0, 0, \ldots, 0)$, then its weight is d^{-1}, and if $|g| = 1$, then its weight is $(kd)^{-1}$. The spectrum of Cayley graphs is well understood; we have that the smallest non-zero eigenvalue of L_H is therefore $\lambda(H) = \frac{1}{k}$. Thus H_G is a $1/k$-expander.

A.2 Anchored games

Here, we prove a lower bound on the second eigenvalue of the $(k-1)$-connection graph of an *anchored* game, and show that it is at least $8k/\alpha^k$. Plugging in this bound into Theorem 5 gives us

$$\mathrm{val}(G^{\otimes n}) \leq \exp\left(-\frac{c\,\alpha^{2k}\,\varepsilon^5\,n}{64\,k^2 \log|\mathcal{A}|}\right).$$

This asymptotically matches the bounds obtained in [1] in terms of the dependence on α and k.

Let us first recall the definition of an anchored game.

▶ **Definition 12** (α-anchored games [1]). Given a k-prover game G, and a parameter $\alpha < 1$ we define the α *anchored* game $G_\perp$ as follows: the referee chooses a question tuple $(x^1, \ldots, x^k)$, according to G, and independently, for every $t \in [k]$, replaces x^t by the anchoring symbol $\perp$ with probability α to get the tuple $(x'^1, \ldots x'^k)$. The new domain is thus $\mathcal{X}'^1 \times \mathcal{X}'^2 \ldots \mathcal{X}'^k$, where $X'^i = \mathcal{X} \cup \{\perp\}$. If any of the x''s are $\perp$, the verifier accepts trivially, otherwise the verifier accepts according to the predicate of the game G.

For convenience, we will denote the α-anchored game itself by G in this section, and its $(k-1)$-connection graph by H_G. We will show the following lemma.

[5] Indeed, by letting d be large enough, we can approximate μ arbitrarily well through discretization and identifying $[d]$ with $\mathcal{X}^t$ for $t = 1, \ldots, k$ in a many-to-one-fashion. Our bounds will not depend on d, so d can be taken to be arbitrarily large.

▶ **Lemma 13.** [6] *$\lambda(H_G) \geq \alpha^k/8k$, when $\alpha < 1/2$.*

In order to prove Lemma 13, we need to make a couple of observations. First, note that the 1-connection graph H_G's vertices can be partitioned into disjoint sets $V_0, V_1, \ldots, V_k$, where V_i has vertices of all question-tuples with exactly i bottom symbols. Thus, V_0 has vertices corresponding to the original question tuples, and $V_k = \{(\perp, \perp, \ldots, \perp)\}$. While V_0 has edges between its own vertices (corresponding to edges in the 1-connection graph of the un-anchored game), all other edges in H_G go between V_i and V_{i+1}.

We will lower bound $\lambda(H_G)$ using the notion of *congestion* in the graph. This technique was first introduced by Diaconis and Strook [5], and improved by Sinclair [12]. The below form can be found in the survey [7, Section 4].

Let us view H_G as an undirected graph[7], with weight function ρ on the edges. Since $\rho(x, y) = \rho(y, x)$ by our definition, this is well-defined. A set of canonical paths in H_G is a set $\mathcal{P}$ of simple paths, one between every ordered pair (x, y) in H_G. The *path congestion parameter* of this set of canonical paths is defined as follows:

$$\zeta(\mathcal{P}) \triangleq \max_{e \in E(H_G)} \frac{1}{\rho(e)} \sum_{p_{xy} \ni e} \rho(x)\rho(y)|p_{xy}|$$

Above, p_{xy} denotes the path from x to y in $\mathcal{P}$, and $|p_{xy}|$ is its length. Intuitively, the numerator in the above equation defines the 'load' on the edge (x, y), while $\rho(x, y)$ can be interpreted as its capacity. Thus, one would naturally expect that if we could find a set of canonical paths with low congestion parameter, the graph must be expanding in some sense. This is formalized in the following theorem:

▶ **Theorem 14** ([12], see also [7, Theorem 4.3]). *For any set of canonical paths $\mathcal{P}$,*

$$\lambda(H_G) \geq \frac{1}{\zeta(\mathcal{P})}$$

We will prove Lemma 13, by choosing a good set of canonical paths in H_G.

Proof of Lemma 13. Consider two vertices x, y in H_G. Let $\Delta(x, y) = \{i_1, \ldots, i_s\} \subseteq [k]$ be the set of (player) indices where the tuples differ, with $i_1 \leq i_2 \leq \ldots i_s$. We will define the canonical path from x to y to be the one obtained by flipping each of $x^{i_1}, \ldots x^{i_s}$ to $\perp$ in order, and then flip these to $y^{i_1}, \ldots y^{i_s}$, but in the reverse order $i_s \to \ldots \to i_1$. Each flip corresponds to moving along an edge in H_G. Call the set of these canonical paths $\mathcal{P}$. The path from x to y in $\mathcal{P}$ is exactly the reverse of the path from y to x.

We will upper bound the congestion through any edge $e = \{u, v\}$ caused by $\mathcal{P}$. If $u, v \in V_0$, then no path in $\mathcal{P}$ passes through this edge, and hence the congestion on e is 0. Suppose that $u \in V_l$, and $v \in V_{l+1}$ for some $l < k$.

We need to find which vertices x would use a canonical path that passes from u to v to reach another vertex. To identify this set, define $B_v \triangleq \{i \in [k] \ : \ v_i = \perp\}$, and similarly $B_u \triangleq \{i \in [k] \ : \ u_i = \perp\}$. Clearly $|B_v| = l + 1, |B_u| = l$, and $B_u \subseteq B_v$. Let us write u as $u = (\perp^l, z_u)$, where the indices are appropriately ordered (with z_u in $\overline{B}_u$).

[6] Although the proof of the lemma can be easily seen to show a bound dependent only on α and k for all $\alpha < 1$, the anchored game definition in [1] sets α to be a constant $< 1/2$. We only state this case, for clarity of exposition and comparison to their result.

[7] On the other hand, if viewed as a directed Markov chain, the transition probability $\Pr[y \mid x]$ for moving from x to y is exactly $\rho(x, y)/\rho(x)$. The stationary distribution on every vertex is $\rho(x)$.

For $0 \leq r \leq l$, a vertex $w \in V_r$ will be said to be in the r-th *shadow* of u (denoted by $S_r(u)$), if:

(a) $w|_{\overline{B}_u} = z_u$, and
(b) If $B_u = \{j_1, \ldots, j_l\}$, with $j_1 \leq \ldots \leq j_l$, then $w_{j_q} \neq \perp$ for every $q > l - r$.

The following Claim is easy to verify:

▶ **Claim 15.** $\rho(S_r(u)) = \Pr_{x\sim\rho}[x^{\overline{B}_u} = z_u] \times \alpha^r(1-\alpha)^{l-r}$

Proof. Any vertex in $S_r(u)$ can be seen to be generated by the verifier in the following way: pick a random question in the original (un-anchored) game conditioned on $x^{\overline{B}_u} = z_u$, then flip $j_1, \ldots j_r$ to $\perp$ (happens with probability α^r), and leave the others unflipped (happens with probability $(1-\alpha)^{k-r}$). The probability of not flipping $\overline{B}_u$ (i.e. $(1-\alpha)^{k-l}$) is accounted for in the distribution ρ of the anchored game. This yields the measure of the set $S_r(u)$ as being the expression given above. ◀

Any path in $\mathcal{P}$ that passes through u will necessarily either originate or end in one of its shadows. The length of any canonical path as defined above is at most $2k$. Hence, the load through the edge (u, v) can be upper bounded as follows (denoting $\Pr_{x\sim\rho}[x^{\overline{B}_u} = z_u]$ by $\Pr[z_u]$ for clarity):

$$\begin{aligned}
\sum_{p_{xy}\ni e} \rho(x)\rho(y)|p_{xy}| &\leq 2k \sum_{p_{xy}\ni e} \rho(x)\rho(y) \\
&\leq 4k \sum_{r=0}^{l} \rho(S_r(x)) \\
&= 4k \sum_{r=0}^{l} \Pr_{x\sim\rho}[x^{B_u} = z_u] \times \alpha^r(1-\alpha)^{l-r} \\
&= 4k(1-\alpha)^l \Pr[z_u] \sum_{r=0}^{l} \left(\frac{\alpha}{1-\alpha}\right)^r \\
&\leq 4k(l+1)(1-\alpha)^l \Pr[z_u] \qquad \ldots \text{ since } \alpha < 1/2 \\
&\leq 8k \Pr[z_u]
\end{aligned}$$

The capacity of edge (u, v) is $\rho(u, v) = \Pr[z_u] \times \alpha^l$. Thus, the congestion along the edge is bounded by

$$\zeta(e) \leq \frac{8k \Pr[z_u]}{\Pr[z_u] \times \alpha^l} = \frac{8k}{\alpha^l}.$$

Hence, the maximum congestion is bounded by $\zeta(\mathcal{P}) \leq \frac{8k}{\alpha^k}$, which yields the lower bound $\lambda(H_G) > \frac{\alpha^k}{8k}$, by invoking Theorem 14. ◀

A.3 Connected games

This follows from the observation that $\lambda(H) \geq \rho_{min}$ when the graph H is connected. The "in particular" statement follows from the definition of the weight function ρ of the $(k-1)$-connection graph: $\mathsf{P}_X(x)$ is simply $1/|S|$, and $\mathsf{P}_{X^t|x^{-t}}(x'^t)$ is also at least $1/|S|$.

Cumulative Space in Black-White Pebbling and Resolution*

Joël Alwen[1], Susanna F. de Rezende[2], Jakob Nordström[3], and Marc Vinyals[4]

1 **IST Austria, Vienna, Austria**
`jalwen@ist.ac.at`
2 **KTH Royal Institute of Technology, Stockholm, Sweden**
`sfdr@kth.se`
3 **KTH Royal Institute of Technology, Stockholm, Sweden**
`jakobn@kth.se`
4 **KTH Royal Institute of Technology, Stockholm, Sweden**
`vinyals@kth.se`

Abstract

We study space complexity and time-space trade-offs with a focus not on peak memory usage but on overall memory consumption throughout the computation. Such a cumulative space measure was introduced for the computational model of parallel black pebbling by [Alwen and Serbinenko 2015] as a tool for obtaining results in cryptography. We consider instead the non-deterministic black-white pebble game and prove optimal cumulative space lower bounds and trade-offs, where in order to minimize pebbling time the space has to remain large during a significant fraction of the pebbling.

We also initiate the study of cumulative space in proof complexity, an area where other space complexity measures have been extensively studied during the last 10–15 years. Using and extending the connection between proof complexity and pebble games in [Ben-Sasson and Nordström 2008, 2011], we obtain several strong cumulative space results for (even parallel versions of) the resolution proof system, and outline some possible future directions of study of this, in our opinion, natural and interesting space measure.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes – Relations among complexity measuresm, F.4.1 Mathematical Logic – Computational logic, F.2.2 Nonnumerical Algorithms and Problems – Complexity of proof procedures

Keywords and phrases pebble game, pebbling, proof complexity, space, cumulative space, clause space, resolution, parallel resolution

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.38

1 Introduction

The time and space complexity measures are at the heart of understanding computation. Unfortunately, there is little we can say about general computation models such as Boolean circuits, let alone Turing machines. But if we allow ourselves to work with simpler models of computation, then we have a better chance at understanding these resources, and in fact there has been impressive progress in restricted models like bounded-depth circuits.

* The second, third and fourth authors were funded by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013) / ERC grant agreement no. 279611. The second author was also supported by Swedish Research Council grants 621-2010-4797 and 621-2012-5645.

© Joël Alwen, Susanna F. de Rezende, Jakob Nordström, and Marc Vinyals;
licensed under Creative Commons License CC-BY

One of the first success stories in this direction are pebble games. The original *(black) pebble game* is played by a single player on a directed acyclic graph (DAG) with a single sink and all vertices having bounded indegree and consists of two simple rules:

1. we can add a pebble to a vertex if all its direct predecessors have pebbles, and
2. we can remove a pebble from a vertex at any time.

The goal of the game is to place a pebble on the sink of the graph. Time is measured as the number of moves to reach this goal, and the space is the maximum number of pebbles needed simultaneously at any point during the pebbling.

Quite surprisingly, this seemingly simple and innocent game can be used to obtain strong results even for general computation models, as it is at the core of the $\mathsf{DTIME}(t) \subseteq \mathsf{SPACE}(t/\log t)$ space upper bound for Turing machines in [33]. Pebbling was first used in [44] to study flowcharts and recursive schemata, and different variants of the game have later been applied to a rich selection of problems in computer science, including register allocation [50], algorithmic time and space trade-offs [17], parallel time [23], communication complexity [47], monotone space complexity [16, 28], cryptography [3, 22], and proof complexity [9, 11, 15] (where it should be emphasized that the above list of references is far from exhaustive). An excellent overview of pebbling up to ca 1980 is given in [46] and another in-depth treatment of some pebbling-related questions can be found in chapter 10 of [48]. Some more recent developments are discussed in the upcoming survey [42].

Let us briefly discuss what is known about space in proof complexity, since this is one of the two topics we are focusing on in this paper. The study of space in proof complexity was initiated in [25], which introduced the *clause space* measure for the well-known resolution proof system, a measure that has subsequently been thoroughly investigated. Informally, the clause space of a resolution proof can be defined as the maximal number of additional clauses – on top of the clauses in the original CNF formula – that a verifier needs to keep in memory at any time while checking the correctness of the proof.[1] While some formulas have proofs requiring only a small, sometimes even just constant, space overhead during verification, other formulas require a linear amount of extra space [1, 8, 25], and as shown in [25] no formulas require more than linear clause space in resolution.

Other papers have studied how space relates to other proof complexity measures. With respect to proof length, which can be viewed as a measure of (nondeterministic) running time, there is a wide range of trade-off results. It has been shown that there are formulas which have both short and space-efficient proofs, but as one of these measures is optimized the other one can blow up to almost worst-case behaviour [10]. Not only this, but there are even formulas where short proofs require more than the worst-case linear space [6, 7]. Yet other papers have studied other space measures such as *total space* [1, 12, 14], measuring the total number of symbols in a proof, and space complexity has also been considered for other proof systems than resolution. We refer the reader to the survey [41] for more details (although, for obvious reasons, it fails to cover the very latest results on total space).

All the space measures discussed above have in common the fact that they refer to the maximum space used at some point in the proof, but they are far from providing a complete picture of space usage during the whole proof. If we only know that a formula has high space complexity, it is not possible to distinguish between a formula that requires large space only at the beginning of the proof, say, and another that requires large space throughout the whole proof. This distinction might not be so important if we are considering the memory requirements of a verifier, since in this case we are chiefly interested in the

[1] Though slightly different from the definition in [25], this is equivalent up to a small additive constant.

maximum. However, it could be relevant for proof search: an algorithm that searches for a proof by producing clauses needs to discard many of them or risk exhausting its available memory. In this case, the difference between needing large space once versus at all times is the difference between making one lucky choice of which clauses to keep in memory versus being lucky all the time.

A similar issue occurs with so-called *memory-hard functions* in the context of cryptography. The idea behind memory-hard functions is that they should require a large amount of memory to evaluate, so that in order to compute such a function for many inputs as a part of a brute-force attack either an infeasible amount of memory is needed or the attack needs to be carried out sequentially. Yet, if the function only requires a large amount of memory during a limited time of the computation, then it is possible to reuse memory for different computations overlapping suitably in time as observed in [2]. Therefore, a more appropriate measure to analyse memory-hard functions is *cumulative space complexity* as introduced in [3], where one measures not the maximum memory consumption but the total memory usage aggregated over the time of the computation.

Although with hindsight this cumulative space complexity measure appears to be a very natural way of quantifying memory usage, it does not seem to have received too much attention in computational complexity theory, and to the best of our knowledge it has not been considered at all in the context of proof complexity. One of the main contributions of this paper is to transfer the concept of cumulative space to proof complexity and to initiate a study of this complexity measure for the resolution proof system.

Pebble games turn out to be a useful tool also for analysing cumulative space. For pebbling strategies cumulative space is straightforwardly defined as the sum over all steps of the pebbling of the number of pebbles on the DAG at each point in time. Thus, in the standard pebble game discussed above any DAG with n vertices can trivially be pebbled in time n and cumulative space $\mathrm{O}(n^2)$ by placing pebbles on all vertices in topological order. Since every vertex needs to be pebbled at some point, a trivial lower bound for the cumulative space is n. However, depending on the intended application one needs to consider other variations of this pebble game as discussed next.

In a proof complexity setting we need to study the *black-white pebble game*, which was introduced in [20] with the objective of modelling nondeterministic computations. Here white pebbles, corresponding to nondeterministic guesses, can be placed at any vertex at any time, but such a white pebble can only be removed from a vertex when all direct predecessors have (black or white) pebbles, corresponding to that the correctness of the nondeterministic guess can be verified.

To model parallel computation in a cryptographic setting, [3] introduced yet another pebble game, namely the *parallel (black) pebble game*. In this game, all the pebbling moves that are legal at some point in time can be performed simultaneously in one single step. This change of rules does not affect the maximal space required to pebble a DAG, but typically changes the pebbling time. Any connected DAG with a single sink requires linear time to pebble sequentially, but for a parallel pebbling it is easy to see that the time required is upper-bounded by the depth of the graph (i.e., the length of a longest path). We remark that an attractive feature of parallel pebbling is that it better captures the difference between maximal and cumulative space. Note that in any sequential pebbling game placing s pebbles requires s time steps, and during the last $s/2$ steps there will be at least $s/2$ pebbles on the DAG. Thus, any pebbling in maximal space s requires cumulative space $\Omega(s^2)$. In contrast, in a parallel pebbling the cumulative space can be small even when the maximal space is large.

1.1 Our Pebbling Contributions

In this paper, we study the cumulative space measure in the context of black-white pebbling. In order to do so, we also extend black-white pebbling to a parallel version. As pebble games go, this is a very powerful model, since it turns out that any DAG can be pebbled with a parallel black-white pebbling in constant time and linear cumulative space. Perhaps somewhat surprisingly, however, it is still possible to prove nontrivial time-space trade-offs. It can be shown that the parallel and sequential versions of black-white pebbling are closely connected (as discussed in more detail later in the paper), and therefore in this overview the exposition is focused on sequential black-white pebbling.

The first question we address is how the large cumulative space can be in the worst case for sequential black-white pebbling. As noted above, a trivial (black-only) pebbling in linear time and space has cumulative space $\mathrm{O}\big(n^2\big)$ for any graph over n vertices. In the other direction, the $\Omega(n/\log n)$ space lower bound in [30] already gives a $\Omega(n^2/\log^2 n)$ cumulative space lower bound for sequential black-white pebbling, as explained above. One cannot get a better cumulative space lower bound by this simple argument from maximal space lower bounds, however, since any DAG of constant indegree can be pebbled in maximal space $\mathrm{O}(n/\log n)$ [33].

We prove that the family of *grate graphs* in [49] require $\Omega(n^2)$ cumulative space for sequential black-white pebbling. This shows that for cumulative space it is not possible to improve on the trivial quadratic upper bound, in contrast to the maximal space measure where it is always possible to save a logarithmic factor from the trivial linear upper bound. This is also different from the parallel black pebble game, where there is a $\mathrm{o}(n^2)$ worst-case upper bound for cumulative space [2] and the best known cumulative space lower bound is $\Omega\big(n^2/\log n\big)$ [4]. In fact, it turns out that the difference between the sequential black-white and parallel black pebble games can be very large. We also prove that (a modified version of) the *butterfly graphs* in [51] require cumulative space $\Omega(n^2/\log n)$ in the sequential black-white pebble game but can be pebbled in linear cumulative space in the parallel black pebble game. Butterfly graphs also show that graphs that require large cumulative space do not necessarily require large maximal space, as they have logarithmic depth and thus can be pebbled in logarithmic space as observed in [33]. We obtain these results by studying the lower bounds on cumulative space in parallel black pebbling in [4] in terms of *depth-robustness* of graphs, and extending these lower bounds to other pebble games and other families of graphs.

Our next set of results concern trade-offs between time and space. Here our starting point is the family of *bit-reversal permutation graphs* studied in [37] which can be pebbled either with 3 pebbles or (as any graph) in linear time, but for which any pebbling in time t and space s must satisfy $t = \Omega(n^2/s^2)$, where as before n is the number of vertices in the graph.

We strengthen this trade-off to cumulative space, proving that pebblings of these graphs in space s require cumulative space $\Omega\big(n^2/s\big)$, which in particular implies that a pebbling in time $\mathrm{O}\big(n^2/s^2\big)$ must use space $\Omega(s)$ not only at some point but most of the time.[2] Furthermore, we establish an unconditional $\Omega\big(n^{3/2}\big)$ cumulative space lower bound, which provides another example of graphs that require (at least somewhat) large cumulative space but can be pebbled in very small (even constant) maximal space. Our proofs of these results work by adapting the *dispersion* technique from [4]. This technique has the advantage that it isolates an abstract combinatorial property of the graph that makes the lower bound argument go through, and this cleaner approach enables us to prove these results not only

[2] Note, importantly, that such a space lower bound is *not* implied by the simple "space s implies cumulative space $\Omega\big(s^2\big)$" argument discussed previously.

for bit-reversal graphs but also for *random permutation graphs* (by showing that these graphs possess the required combinatorial property with high probability). To the best of our knowledge no trade-offs (even non-cumulative ones) were known for such graphs before for any flavour of the pebble game.

Finally, we consider a very concrete, extremal question regarding pebbling time-space trade-offs. It is an easy observation that any sequential black-white pebbling in constant space s can be carried out in time $\mathrm{O}(n^s)$, since there are only $\sum_{k=0}^{s} 2^k \binom{n}{k}$ possible different configurations of s pebbles in the graph, and no configuration repeats in a pebbling (or else the intermediate moves can be removed). In fact, a bit more thought reveals that this time bound can be sharpened to $\mathrm{O}(n^{s-1})$, since every configuration in space s is immediately followed by a pebble removal, and so we only need to consider distinct configurations of $s-1$ pebbles. It is a natural question whether this simple counting argument is in fact tight, so that there are graphs that can be pebbled in space s but where any such pebbling requires time $\Omega(n^{s-1})$.

For pebbling space $s = 3$, the minimum space in which any nontrivial pebbling strategy is possible, the bit-reversal graphs in [37] discussed above show that the answer to this question is affirmative. It is not hard to see that by stacking $s-2$ bit-reversal DAGs on top of one another, identifying the top layer in one graph with the bottom layer in the graph above, one obtains graphs that are pebblable in space s but where the obvious pebbling strategy achieving this bound requires time $\mathrm{O}(n^{s-1})$. We prove that this trivial upper bound is indeed asymptotically tight for any constant s.

1.2 Our Proof Complexity Contributions

Turning now to proof complexity, we consider the main contribution of our paper to be that we initiate the study of the cumulative space measure. While the concept of cumulative space seems to be as natural as maximal space, we are not aware of it having been studied in the context of proof complexity before. As was the case for the first papers on (maximal) space complexity in resolution [25], in this first paper on cumulative space in proof complexity we focus on the resolution proof system.

An immediate observation is that proof length is always a lower bound on cumulative space, and so exponential lower bounds on proof length – as shown for resolution in [18, 32, 52] and many later papers – trivially imply exponential lower bounds on cumulative space. Therefore, it seems that the cumulative space measure will be of independent interest mostly for formulas which have reasonably short proofs. An obvious candidate family to study are pebbling formulas [11], which have proofs in linear length, but which exhibit a rich variety of properties with respect to space complexity depending on the underlying graphs in terms of which they are defined.

However, we also need to decide on an appropriate model of the resolution proof system in which to study cumulative space. In the context of pebbling we concluded that cumulative space makes most sense for parallel versions of the pebble games, and so it is natural to ask whether one should consider a parallel version of resolution when studying cumulative clause space. It is not hard to argue that such a parallel model of resolution could be interesting in its own right, since it might be useful as a tool to analyse attempts to parallelize state-of-the-art SAT solvers using so-called *conflict-driven clause learning (CDCL)* [5, 38].

We define and study several different versions of the resolution proof systems with varying degrees of parallelity. The running time of parallel CDCL solvers has previously been analysed using resolution depth and the related *conflict resolution depth* and *schedule makespan* measures introduced in [36], and our models of parallel resolution allow us to reason about space in addition to time.

Similarly to what is the case for pebble games, our most general model of parallel resolution, where clauses can be inferred not just by syntactic application of the resolution rule but by semantic inference, is extremely powerful, so much so that it can deal with any formula in a constant number of steps and linear space. Since we can establish a tight relation between space and parallel speedup also for resolution, however, we can still obtain lower bounds when the maximal space is limited.

Studying pebbling formulas in these different models of resolution, and revisiting the reductions between resolution and pebble games in [9, 10], we can translate the pebbling results in Section 1.1 to results for the resolution proof system. Summarizing very briefly, we exhibit different formulas that have

- proofs in linear length but require quadratic cumulative space,
- proofs in logarithmic space but require $\Omega(n^2/\log n)$ cumulative space, and
- trade-offs between proof length and cumulative space.

1.3 Paper Outline

The rest of this paper is organized as follows. In Section 2 we present a more detailed overview of our pebbling results, introducing formal definitions of the pebble games and measures discussed above, and we give an analogous overview for resolution in Section 3. The reader is referred to the upcoming full-length version for all missing proofs. We conclude in Section 4 with a discussion of possible directions for future research.

2 Pebbling Results Overview

Let us start our pebbling overview by giving formal definitions of the basic concepts.

2.1 Definition of Pebble Games and Basic Properties

We say that a directed acyclic graph (DAG) $G = (V, E)$ with $|V| = n$ has *size* n. A vertex $v \in V$ has *indegree* δ if it has δ incoming edges $\{(u_1, v), \ldots, (u_\delta, v)\} \subseteq E$, $u_i \neq u_j$ for $i \neq j$, and we say that G has indegree δ if the maximum indegree of any vertex of G is δ. A vertex with no incoming edges is called a *source* and a vertex with no outgoing edges is called a *sink*. We say that a vertex u is a *predecessor* of a vertex v if there exists a directed path from u to v; moreover, if this path consists of only one edge then u is a *direct predecessor* of v. We denote by $\mathsf{parents}(v)$ the set of all direct predecessors of v. For technical reasons, it will sometimes be convenient to allow paths of length 0 in the definition above, so that a vertex can be a predecessor of itself. We will sometimes consider graphs obtained from other graphs by removing subsets of vertices, and for $U \subseteq V$ we write $G - U = \big(V \setminus U, E \setminus ((U \times V) \cup (V \times U))\big)$ to denote the DAG obtained from G by removing the vertices in U and all edges incident to U.

To get a unified description of all flavours of the pebble game discussed in Section 1, it is convenient to define pebbling as follows.

▶ **Definition 1** (Pebble games)**.** Let $G = (V, E)$ be a DAG with a unique sink vertex z. The black-white pebble game on G is the following one-player game. At any time i, we have a *black-white pebbling configuration* $\mathbb{P}_i = (B_i, W_i)$ of black pebbles B_i and white pebbles W_i on the vertices of G, at most one pebble per vertex. The rules of how a pebble configuration $\mathbb{P}_{i-1} = (B_{i-1}, W_{i-1})$ can be changed to $\mathbb{P}_i = (B_i, W_i)$ are as follows:

1. A black pebble may be placed on a vertex v only if all immediate predecessors of v are covered by pebbles in both $\mathbb{P}_{i-1}$ and $\mathbb{P}_i$, i.e.,

   $$v \in (B_i \setminus B_{i-1}) \;\Rightarrow\; \mathsf{parents}(v) \subseteq \mathbb{P}_{i-1} \cap \mathbb{P}_i \;\;.$$

   Note that, in particular, a black pebble can always be placed on a source vertex.
2. A black pebble on any vertex v in $\mathbb{P}_{i-1}$ can be removed in $\mathbb{P}_i$.
3. A white pebble can be placed on any vertex v in $\mathbb{P}_i$.
4. A white pebble on a vertex v in $\mathbb{P}_{i-1}$ may be removed in $\mathbb{P}_i$ only if all immediate predecessors of v are covered by pebbles in both $\mathbb{P}_{i-1}$ and $\mathbb{P}_i$, i.e.,

   $$v \in (W_{i-1} \setminus W_i) \;\Rightarrow\; \mathsf{parents}(v) \subseteq \mathbb{P}_{i-1} \cap \mathbb{P}_i \;\;.$$

   In particular, a white pebble can always be removed from a source vertex.

A *legal pebbling* $\mathcal{P}$ of G is a sequence $\mathcal{P} = (\mathbb{P}_0, \ldots, \mathbb{P}_t)$ where every configuration $\mathbb{P}_i$ can be obtained from $\mathbb{P}_{i-1}$ using the rules 1–4. A *complete pebbling* $\mathcal{P} = (\mathbb{P}_0, \ldots, \mathbb{P}_t)$ is a legal pebbling where $\mathbb{P}_0 = \mathbb{P}_t = (\emptyset, \emptyset)$ and $z \in \bigcup_{i=0}^{t}(B_i \cup W_i)$ (i.e., the sink is pebbled at some point).

A *black pebbling* is a pebbling where $W_i = \emptyset$ for all $i \in [t]$. A pebbling is *sequential* if only a single application of a single rule 1–4 is used to get from from $\mathbb{P}_{i-1}$ to $\mathbb{P}_i$ for all $i \in [t]$. In a *(fully) parallel* pebbling an arbitrary number of applications of the rules 1–4 can be made to $\mathbb{P}_{i-1}$ to obtain $\mathbb{P}_i$ (but note that all pebble placements and removals have to be legal with respect to $\mathbb{P}_{i-1}$, and cannot make use of any pebble placements or removals made in parallel). Finally, we will also consider *parallel-black sequential-white* pebblings, which allows parallel applications of black pebble rules 1–2 to $\mathbb{P}_{i-1}$ to obtain $\mathbb{P}_i$, but only a single application of the white pebble rules 3–4. Note that, in the parallel setting, a simultaneous application of rules 1 and 4 on a same vertex replaces a white pebble by a black one.

The *time* of a pebbling $\mathcal{P} = (\mathbb{P}_0, \ldots, \mathbb{P}_t)$ is $t(\mathcal{P}) = t$; the (maximal) *space* is $s(\mathcal{P}) = s = \max_{i\in[t]}|B_i| + |W_i|$; and the *cumulative space* is $c(\mathcal{P}) = c = \sum_{i\in[t]}|B_i| + |W_i|$ (where we observe that $c \leq st$).

Parallel black pebbling was introduced in [3], where it was pointed out that for certain graphs parallel pebblings can be much more efficient than sequential, while for others they cannot do any better. For example, if we are considering time-space tradeoffs, any sequential black pebbling in space s and time t of the bit-reversal graph must satisfy $st = \Omega(n^2)$ [37], while in the parallel black game one can pebble such graphs in linear time and space $\mathrm{O}(\sqrt{n})$ [3]. In contrast, it was shown in [4] that there are graphs that can be pebbled sequentially in space s and time t satisfying $st = \mathrm{O}(n^2/\log n)$, but where these graphs even in the parallel model require not only $st = \Omega(n^2/\log n)$ but also cumulative space $\Omega(n^2/\log n)$.

Unlike the case of the black pebble game, we show that time and space in the black-white sequential and parallel games are closely related. Up to constant factors, it holds that if a parallel black-white pebbling $\mathcal{P}$ has maximal space s, then it is possible to save a factor s, but not more than a factor s, in time compared to a sequential black-white pebbling in the same space s.

▶ **Observation 2.** *Let $\mathcal{P}$ be a parallel black-white pebbling of a DAG G in time t, space s, and cumulative space c. Then there is a sequential black-white pebbling of G in time $2ts$, space $2s$, and cumulative space cs.*

Proof. Each parallel move places at most s pebbles and removes at most s pebbles, therefore we can simulate it by $2s$ sequential moves (making the pebble placements first, to make sure that these moves remain legal). ◀

▶ **Lemma 3.** *Let $\mathcal{P}$ be a sequential black-white pebbling of G in time t, space s, and cumulative space c, and let k be a positive integer. Then there is a parallel black-white pebbling of G in time $3\lceil t/k \rceil$, space $s + \lceil k/2 \rceil$, and cumulative space $3\lceil c/k \rceil + t$.*

Proof. We divide $\mathcal{P}$ into $\lceil t/k \rceil$ intervals of (at most) k moves. We reorder the pebbling moves within each of these intervals so that we do all placements first and removals afterwards. This is still essentially a valid pebbling, because each configuration is a superset of the corresponding configuration in $\mathcal{P}$, except that we can possibly have vertices temporarily covered by several pebbles. The space usage in any intermediate configuration increases to at most $s + \lceil k/2 \rceil$. We then collapse each subsequence into one parallel placement of white pebbles, one step replacing white pebbles with black pebbles as needed, and one parallel removal of black pebbles (this allows us to make all black pebble placements in parallel even though later black pebbles might be dependent on earlier pebble placements in the sequential pebbling). This decreases the time to $3\lceil t/k \rceil$.

To bound the cumulative space, note that if there is a configuration $\mathbb{P}_i$ in a sequential interval that has space s_i, then the corresponding three parallel configurations have aggregate space at most $3s_i + 2x_j$, where x_j is the number of placements in that interval. Now consider a partition of the sequential pebbling into k subsequences $\mathbb{P}_i, \mathbb{P}_{i+k}, \ldots, \mathbb{P}_{i+k(\lceil t/k \rceil - 1)}$ of t/k configurations, evenly spaced, starting at $i \in [1, k]$. By an averaging argument, at least one of these k subsequences has a cumulative space of at most $\lfloor c/k \rfloor$. Hence the total cumulative space is at most $\sum_{j \in [t/k-1]} (3s_{i+j} + 2x_{i+j}) = 3\sum_{j \in [t/k-1]} s_{i+j} + 2\sum_{j \in [t/k-1]} x_j \geq 3c/k + 2t/2$. ◀

Observe that when $k = \Theta(s)$ the cumulative space in Lemma 3 is dominated by the term t, so we only save a factor s in cumulative space when the sequential pebbling has cumulative space $c = \Theta(st)$. Since the graphs we will discuss in what follows have cumulative space lower bounds of this form, studying the sequential game already gives us all the information we want about the parallel game.

▶ **Corollary 4.** *Let $\mathcal{P}$ be a black-white pebbling of G in time t and space s. Then there is a parallel black-white pebbling of G in time $\lceil t/2s \rceil$, space $4s$, and cumulative space $2t$.*

2.2 Robustness and High Cumulative Space Complexity

We proceed to define the concept of *depth-robustness* of graphs, which is inspired by [24, 45] and which will be central to our work.

▶ **Definition 5** ($\mathcal{G}$ -robustness)**.** Let $\mathcal{G}$ be a family of DAGs and let $e, d \in \mathbb{N}^+$ be positive integers. We say that a DAG $G = (V, E)$ is (e, d)-*$\mathcal{G}$-robust* if for every subset of vertices $U \subseteq V$ of size at most e it holds that $G - U$ contains a subgraph $H \in \mathcal{G}$ of size at least d.

When $\mathcal{G}$ is the class of directed paths, then we say that G is *depth-robust*, and when $\mathcal{G}$ is the class of DAGs with one sink the DAG G is said to be *predecessor-robust*.[3]

For our pebbling lower bounds we are interested in graphs with very high robustness, i.e., for as large values of e and d as possible. Depth-robustness was first studied by Erdős, Graham and Szemerédi [24] who showed how to construct DAGs with indegree $\Theta(\log(n))$ possessing $(\Omega(n), \Omega(n))$-depth-robustness. However, in our applications it is important that

[3] This choice of terminology is inspired by [45], which discusses the dual notions of "depth-separators" and "predecessor-separators."

the graphs have *constant* indegree. Valiant [53] showed that for constant indegree and linear depth the best we can hope for is $(\mathrm{O}(n/\log n), \mathrm{O}(n))$-depth-robustness. Fortunately for us, it was shown in [4, 45] that such extremal $(\Theta(n/\log n), \Theta(n))$-depth-robust graphs do exist. Conversely, if we want constant indegree with the parameter e linear in the graph size, then $(\epsilon n, n^{1-\epsilon})$-depth-robustness is the best we can hope for [53]. In [49] a family of constant-indegree $(\Theta(n), \Theta(n^{1-\epsilon}))$-depth-robust graphs were presented.

The connection between depth-robustness and cumulative space was made in [4], where it was shown that an (e, d)-depth-robustness graph requires parallel black cumulative space at least ed. In this work, we give a more general theorem of this form for the case of $\mathcal{G}$-robustness. We then use this theorem to obtain the following lower bounds for depth-robust and predecessor-robust graphs.

▶ **Lemma 6.** *If G is an (e, d)-depth-robust DAG, then G requires sequential black-white cumulative space at least ed, and parallel-black sequential-white cumulative space at least $e\sqrt{d}$.*

▶ **Lemma 7.** *If G is an (e, d)-predecessor-robust DAG, then G requires black-white cumulative space at least ed.*

Focusing on the range of parameters discussed above, we can see that, it follows from Lemmas 6 and 7 that a $(\Theta(n/\log n), \Theta(n))$-depth-robust graph has sequential black-white cumulative space complexity $\Omega(n^2/\log n)$ and parallel-black sequential-white pebbling space complexity $\Omega(n^{3/2}/\log n)$.

A class of DAGs that are predecessor-robust are *grates* – graphs with n' sources and n' sinks such that after the removal of an arbitrary set of kn' vertices (for some constant k) there are still a linear number of sources and sinks that are all pairwise connected. *Butterfly graphs* [51] are grates with $n = n' \log n'$ vertices that are $(\Theta(n/\log n), \Theta(n/\log n))$-predecessor-robust. Moreover, it is not hard to show that if we append n' single-sink DAGs of size $\log n'$, one to each source of the butterfly graph, the resulting graph is $(\Theta(n/\log n), \Theta(n))$-predecessor-robust. This implies that these graphs require cumulative space $\Omega(n^2/\log n)$. Note that butterfly graphs (also in the modified version just described) can be pebbled with $\mathrm{O}(\log n)$ pebbles (since the graphs have depth $\mathrm{O}(\log n)$), and thus it is not the case that high cumulative space implies large maximal space.

It has been established that extremal depth-robustness is both a necessary [2] and sufficient [4] condition to have high cumulative space in the parallel black game. In particular, using the fact that no graph of size n with constant indegree is $(\omega(n/\log n), \Theta(n))$-depth-robust, it was shown in [2] that in the parallel black pebble game, for any constant $\epsilon > 0$, any such graph has cumulative space complexity $\mathrm{o}(n^2/\log^{1-\epsilon} n)$. A natural question is if this also holds for black-white pebbling. We show that this is not the case: there are graphs that have maximum cumulative space complexity $\Omega(n^2)$ in the black-white pebble game. This follows from Lemma 7 and the existence of grates of size linear in the number of sources and sinks [49].

2.3 Dispersion and Cumulative Space Trade-Offs

Another property of graphs that is important in the current paper is *dispersion*. This notion was used in [4] to obtain another condition ensuring high parallel black cumulative space complexity. We define two similar concepts and then use them to obtain cumulative space trade-offs. The results we get are for two classes of *permutation graphs* – graphs that consist two ordered paths of vertices $1, 2, \ldots, n$, where in addition an edge is added from each vertex i in the first path to its image under some specified permutation σ in the second path.

A family of permutations that will be of particular interest to us are the so-called *bit-reversal permutations*, which are defined for $n = 2^m$ and which simply reverse the binary representations of numbers. That is, if $j = (b_1 \cdots b_m)_{(2)}$, then the bit-reversal permutation σ sends j to $\sigma(j) = (b_m \cdots b_1)_{(2)}$. It was previously known [37] that any sequential black-white pebbling of a bit-reversal permutation graph on $2n$ vertices in time t and space s satisfies $st = \Omega\big(n^2/s\big)$. Moreover, it was shown in [37] that this is tight up to constant factors and that there is a black-white pebbling in time t and space s such that $st = \mathrm{O}\big(n^{3/2}\big)$.

We observe that while bit-reversal graphs are *not* $(2\sqrt{n}, 2\sqrt{n})$-depth-robust, they can be shown to be $(\sqrt{n}, n)$-predecessor-robust. Therefore, in constrast to [4], where it was not possible to establish a parallel black cumulative space lower bound of $n^{3/2}$ using depth-robustness, we are able to obtain a black-white cumulative space lower bound of $n^{3/2}$ using predecessor-robustness.

Our reason for studying dispersion properties of bit-reversal graphs is to characterize how cumulative space increases when space decreases. We show that the time-space trade-off in [37] can be strengthened to a cumulative space trade-off. Our result implies that if $\mathcal{P}$ is a sequential black-white pebbling of a bit-reversal graph in space s and time n^2/s^2, then it needs to use space s not only at some point of the pebbling, but during a large part of the time.

An advantage of our approach is that we identify a general property of graphs that imply cumulative space trade-offs, so that the task of establishing a trade-off reduces to proving that the graph has this desired property. As a consequence of this simplification, we are able to prove the same kind of trade-off results not only for bit-reversal graphs but also for random permutation graphs, a class of graphs for which it seems nothing was known before.

▶ **Theorem 8.** *If G is a random permutation graph, then it holds asymptotically almost surely that in the sequential black-white pebble game G requires cumulative space $\Omega(n^{3/2})$ and any pebbling $\mathcal{P}$ of G in maximal space s has cumulative space $\Omega(n^2/s)$.*

2.4 Pebblings in Small Space Can Require Maximum Length

Let us finally consider the question of how long a shortest sequential pebbling of a graph can be given constraints on the maximal pebbling space. Without loss of generality, a black pebbling in space s takes time at most $\binom{n}{\leq s} \leq n^s$, simply because there is no need to repeat any pebble configuration. A moment of thought reveals that in fact we get the upper bound $\binom{n}{s-1} + \binom{n}{\leq s-1} \leq n^{s-1}$, since every configuration in maximal space s is followed by an erasure yielding a space-$(s-1)$ configuration, and these configurations also do not repeat. For black-white pebbling the upper bound becomes $2^{s-1}\big(\binom{n}{s-1} + \binom{n}{\leq s-1}\big) \leq 2^{s-1}n^{s-1}$.

As discussed in the introduction, it can be read off from [37] that for space-3 pebblings the $\mathrm{O}\big(n^2\big)$ upper bound is tight up to constant factors – bit-reversal DAGs are examples of graphs for which pebblings in optimal space 3, or indeed any constant space, require quadratic time. We extend this result to any $s = \mathrm{O}(1)$ by exhibiting graphs that can be pebbled in space s but where any such pebbling requires time $\Omega\big(n^{s-1}\big)$. We do this by generalizing permutation graphs to multiple layers, where we have k directed path graphs of length n and $k-1$ layers of permutations between the vertices $1, 2, \ldots, n$ in consecutive paths (so that the permutation graphs considered in [37] are 2-layer bit-reversal graphs with paths of length n). We state two theorems below for the black and black-white sequential pebble games, and just as for the 2-layer graphs in [37] our bounds can be stated not just for minimal space but also an arbitrary space parameter s greater than this minimum.

▶ **Theorem 9.** *Let G be a k-layer bit-reversal graph with paths of length n. Then for any s such that $k+1 \leq s \leq \sqrt{n}$ there exists a sequential black pebbling of G in space s and time $\mathrm{O}(n^k/s^{2k-3})$. Furthermore, every sequential black pebbling of G in space s requires time $\Omega(n^k/s^{2k-3})$.*

▶ **Theorem 10.** *Let G be a k-layer bit-reversal graph with paths of length n. Then for any s such that $k+1 \leq s \leq \sqrt{n}$ there exists a sequential black-white pebbling of G in space s and time $\mathrm{O}(n^k/s^{2k-2})$. Furthermore, every sequential black-white pebbling of G in space s, requires time $\Omega(n^k/s^{2k-2})$.*

Our proofs of these results are inspired by the reasoning in [37] for 2-layer permutation graphs, but we also need to overcome some new challenges. The essence of the argument is that in order to place a pebble on the jth layer we need to do some work on the preceding layer. If we only have two layers the argument ends here, but when we want to apply the argument recursively we need to be more careful. Indeed, placing pebbles on the $(j-1)$st layer will now require placing more pebbles on the $(j-2)$nd layer, but if we choose the order in which we do the pebble placements wisely, we may be able to reuse part of the work in the $(j-2)$nd layer for several pebble placements in the $(j-1)$st layer. We are able to find a strategy to exploit this insight and obtain optimal upper bounds, but also to make the lower bound argument resilient enough to get asymptotically matching lower bounds.

3 Cumulative Space for the Resolution Proof System

We now proceed to describe in more detail the proof complexity results in our paper. We start this section by a brief review of some standard proof complexity preliminaries, after which we discuss how to refine the definition of the resolution proof system to be able to make meaningful and precise claims about maximal space and cumulative space. This then allows us to make the connection to the pebbling results in Section 2 and what proof complexity implications they have.

A *literal* over a Boolean variable x is either x itself (a *positive literal*) or its negation $\overline{x}$ (a *negative literal*). A *clause* $C = a_1 \vee \cdots \vee a_k$ is a disjunction of literals a_i over pairwise disjoint variables. A *k-clause* is a clause that contains at most k literals. A *CNF formula* $F = C_1 \wedge \cdots \wedge C_m$ is a conjunction of clauses and a *k-CNF formula* is a CNF formula consisting of k-clauses. We think of clauses and CNF formulas as sets: order is irrelevant and there are no repetitions.

The standard definition of a *resolution refutation* $\pi : F \vdash \bot$ of an unsatisfiable CNF formula F – or a *resolution proof* for (the unsatisfiability of) F – is as an ordered sequence of clauses $\pi = (D_1, \ldots, D_t)$ such that $D_t = \bot$ is the empty clause containing no literals, and each clause D_i, $i \in [t]$, is either an *axiom* $D_i \in F$ or is derived from clauses D_j and D_k, $j, k < i$, by the *resolution rule*

$$\frac{B \vee x \qquad C \vee \overline{x}}{B \vee C}\ , \tag{1}$$

where we refer to $B \vee C$ as the *resolvent over x* of $B \vee x$ and $C \vee \overline{x}$.

In order to study space in general, and cumulative space in particular, we refine the above definition into a family of proof systems as follows.

▶ **Definition 11** (Resolution)**.** A resolution refutation $\pi : F \vdash \bot$ of a CNF formula F is a sequence of *configurations*, or sets of clauses, $\pi = (\mathbb{C}_0, \ldots, \mathbb{C}_t)$ such that $\mathbb{C}_0 = \emptyset$, $\bot \in \mathbb{C}_t$, and for all $i \in [t]$ we obtain $\mathbb{C}_i$ from $\mathbb{C}_{i-1}$ by applying exactly one of the following type of rules:

Axiom download Add $A \in F$.
Inference Add D derived from clauses in $\mathbb{C}_{i-1}$.
Erasure Remove clauses from $\mathbb{C}_{i-1}$.
We say that a refutation is (a) *sequential* if at every time step we apply the chosen rule exactly once; (b) *inference-parallel* if only one clause can be downloaded but the inference rule can be applied an arbitrary number of times (but always deriving from $\mathbb{C}_{i-1}$); and (c) *fully parallel* (or just *parallel*) if both axiom download and inference rules can be applied an arbitrary number of times (but note that we cannot mix applications of different rules in the same step). Furthermore, a refutation is said to be (1) *syntactic* if inferences use the resolution rule (1) and (2) *semantic* if instead any clause D such that $\mathbb{C}_{i-1} \models D$ can be inferred immediately.

The *length* of a resolution refutation π is the number of derivation steps t and the *size* is the total number of clauses introduced in downloads and inference steps (counted with repetitions). The *maximal (clause) space*, or just *space*, of π is $\max\{|\mathbb{C}_i| : \mathbb{C}_i \in \pi\}$ and the *cumulative (clause) space* is $\sum_{\mathbb{C}_i \in \pi} |\mathbb{C}_i|$.

Note that Definition 11 yields a total of six different flavours of resolution 1(a)–2(c) depending on the amount of parallelism and on whether inferences are syntactic or semantic. In what follows, we will discuss our motivation for considering these different models and what we can say about them.

A first, general comment is that from a proof complexity point of view we are mainly interested in *syntactic* versions of the proof systems in Definition 11. Strictly speaking, the *semantic* versions are not even propositional proof system in the sense of Cook and Reckhow [19], since we do not know how to verify semantic implications in polynomial time. In any semantic system we can download all axioms in the formula and then derive contradiction in a single inference step, and efficiently verifying such an inference means solving Sat in polynomial time. However, most results on (clause) space in the proof complexity literature actually hold in the stronger semantic setting. For maximal space this is not so surprising, since the semantic and syntactic space measures are within a constant factor of each other [1], but even for trade-offs one tends to get results in the semantic setting for free (with the notable exceptions of [6, 7]).

Syntactic sequential resolution is the standard definition discussed at the beginning of this section (and note that for this version of resolution the length and size measures are essentially the same). A somewhat unsatisfactory feature of this model is that (analogously to what is the case for pebbling) a maximal space lower bound s immediately implies a cumulative space lower bound $\Omega(s^2)$. The reason is completely analogous: since we can only infer one new clause per time step, during the $s/2$ time steps before reaching space s we must have had at least $s/2$ clauses in memory. It turns out, however, that we can actually beat this lower bound in certain settings, and we also remark that cumulative length-space trade-offs do not necessarily follow from such trivial arguments and so make sense even for syntactic sequential resolution.

By allowing parallel application of inference steps we want to try to get away from cumulative space lower bounds that hold only for the trivial reason just discussed. In *syntactic inference-parallel resolution* we therefore allow clauses to be derived in parallel. As it turns out, anything we are currently able to prove for this model we can also establish for the stronger *semantic inference-parallel resolution* system.

We can also go in the other direction from the syntactic sequential model and introduce a parallelism of sorts by studying *semantic sequential resolution*. As already alluded to, this is a very powerful system since any formula can be refuted in linear size and space by

downloading all its axioms in a linear number of steps and then deriving contradiction in just one semantic inference step, but nevertheless the space lower bounds and length-space trade-offs in [9, 10] hold in this model, and can in fact be verified to hold even for semantic inference-parallel resolution.

The most challenging models in terms of lower bounds are the fully parallel ones. *Syntactic parallel resolution* could be viewed as a potentially interesting model for proving lower bounds on parallel SAT solvers using conflict-driven clause learning, where one could imagine an arbitrarily large number of solvers producing resolvents in parallel and having perfect access to shared memory. It is not hard to see that if a standard resolution proof is represented as a DAG in the natural way, then syntactic parallel length, which would be a proxy for execution time, is just the depth of this DAG.

In the semantic model, adding also parallel axiom downloads makes the proof system exceptionally powerful, since now any formula can be refuted in constant length 2, linear size, and linear cumulative space. This seems a bit too strong to be really interesting (and can be viewed as a reason for preferring the inference-parallel version described previously). However, we shall see that even for semantic fully parallel resolution it is still possible to obtain nontrivial trade-off results if the maximal (non-cumulative) space is bounded.

Moving on from this philosophical discourse to a more concrete discussion of results, we note that most of the proof complexity consequences we derive from the pebbling results in Section 2 are for semantic inference-parallel resolution, and thus hold for all models above except the fully parallel ones. We start by reporting a disappointing fact, however: even in semantic inference-parallel resolution we have the problem that cumulative space is at least maximal space squared.

▶ **Lemma 12.** *If F requires maximal space s in semantic inference-parallel resolution, then any semantic inference-parallel refutation of F has cumulative space $\Omega(s^2)$.*

Proof. For simplicity let us think of each step in a semantic inference-parallel resolution refutation as being either an inference-plus-erasure step or a download step. Clearly, this can only affect the clause space measure by a factor 2.

An inference-plus-erasure step can be seen as a compression operation. Since the proof system is semantic, we only care about the information contained in a configuration, and since an inference step cannot increase the information but only add explicitly clauses that are already implied by the configuration, there is no need to add any extra clauses on top of the minimum amount needed to encode the semantic information we want the proof to maintain at this point. Therefore, without loss of generality the number of clauses only increases at download steps, and since these are sequential we can conclude that the number of clauses increases by at most 1 at every step.

But this means that we can apply the same argument as for syntactic sequential resolution above: during the $s/2$ time steps preceding a space-s configuration we must have at least $s/2$ clauses in memory, and hence a cumulative lower bound $\Omega(s^2)$ follows. ◀

It is important to note, though, that Lemma 12 has no implications for cumulative space trade-offs for formulas where the maximal space complexity is at most $O(\sqrt{N})$ measured in the formula size N, since in this setting the max-space-squared argument only implies a trivial $\Omega(N)$ cumulative space lower bound, and we present such trade-off results below that do not follow from Lemma 12. We also report results that asymptotically beat the maximal-space-squared lower bound for cumulative space.

In order to obtain these results, we need to review how our cumulative pebbling results in Section 2 can be translated to claims about resolution refutations of so-called *XORified*

pebbling formulas. We will be very brief here, since all that needs to be done is to read the pebbling-to-resolution reductions in [10] and verify that the proofs work not only for semantic sequential resolution but also for semantic inference-parallel resolution. We just state the reduction that we need below, since we can use it in a completely black-box fashion without knowing any details about what these formulas are. The interested reader is referred to [10] for the missing details.[4]

▶ **Theorem 13** (by the proof of Theorem 2.1 in [10]). *Let π be a semantic inference-parallel resolution refutation of a XORified pebbling formula $Peb_G[\oplus]$ in length L, maximal space s, and cumulative clause space c. Then there is a sequential black-white pebbling of the underlying DAG G in time L, space s, and cumulative space c.*

Analogously to what is the case in [10], the generic reduction in Theorem 13 can now be applied to a multitude of different graph families with different pebbling properties to yield CNF formulas with the same properties in resolution. Below we just give a sample of such results that we find particularly interesting.

For maximal space it is known that formulas refutable in linear size $\mathrm{O}(N)$ never require space more than $\mathrm{O}(N/\log N)$. For cumulative space the lower bound can be truly quadratic, however, beating the max-space-squared bound in Lemma 12 by a factor $\log^2 N$.

▶ **Theorem 14.** *There is a family of 6-CNF formulas $\{F_N\}_{N\in\mathbb{N}^+}$ of size $\Theta(N)$ that have syntactic sequential resolution refutations in size $\mathrm{O}(N)$, and hence also in maximal clause space $\mathrm{O}(N/\log N)$, but for which any semantic inference-parallel refutations require cumulative clause space $\Omega(N^2)$.*

This theorem follows from studying pebbling formulas defined in terms of *grate graphs* as in [49] and using that the high predecessor-robustness of these graphs imply strong lower bounds on cumulative space as stated in Section 2.

A natural question is what cumulative space tells us about maximal space, and in particular whether high cumulative space complexity implies that the maximal space complexity must also be large. This might sound intuitively plausible, but turns out to be false in a very strong sense.

▶ **Theorem 15.** *There is a family of 6-CNF formulas $\{F_N\}_{N\in\mathbb{N}^+}$ of size $\Theta(N)$ that can be refuted in syntactic sequential resolution in size $\mathrm{O}(N)$ and also in maximal clause space $\mathrm{O}(\log N)$, but for which any semantic inference-parallel refutations require cumulative clause space $\Omega(N^2/\log N)$.*

Here the graphs we need are surprisingly simple, namely butterfly graphs. They again have high predecessor-robustness, but since they are shallow the pebbling formulas generated from them have refutations in small maximal space.

Finally, we turn to the question of length-space trade-offs. We remark that in a cumulative space setting formulas for which small-space proofs require superpolynomial length, as in the strongest results in [10, 7, 6], are not too interesting, since length is trivially a lower bound on cumulative space. Rather, we focus on formulas for which small-space proofs incur only a

[4] It might be worth noting, though, that just as in [10] our results hold not only for pebbling formulas substituted with exclusive or – substitution with any so-called *non-authoritarian* (or *robust*) function that can never be fixed by restricting any single variable to some value works fine. Binary exclusive or is just the simplest example of such a function, whereas standard or is a simple non-example since setting a single variable to true fixes the value of the function to true.

polynomial blow-up in proof length. Can we find such formulas for which it holds not only that short proofs must have large maximal space s, but where such short proofs must be memory-intensive in that this amount of space s must be used essentially throughout the whole proof? The answer to this question is yes, and one example are pebbling formulas over the bitreversal permutation graphs studied in [37]. The next theorem follows by combining the reduction in Theorem 13 with the fact that bitreversal graphs are dispersed as stated in Section 2.

▶ **Theorem 16.** *There is a family of 6-CNF formulas $\{F_N\}_{N\in\mathbb{N}^+}$ of size $\Theta(N)$ such that for any $s = \mathrm{O}(\sqrt{N})$ the formula F_N has a syntactic sequential resolution refutation in size $\mathrm{O}(N^2/s^2)$ and maximal clause space $\mathrm{O}(s)$, but any semantic inference-parallel refutation of F_N in maximal space s requires cumulative clause space $\Omega(N^2/s)$.*

In particular, a proof in maximal space s has length $\Omega(N^2/s^2)$, and if furthermore the proof has length $\mathrm{O}(N^2/s^2)$, then $\Omega(N^2/s^2)$ of the configurations have space $\Omega(s)$. Hence, these formulas have syntactic sequential resolution refutations in simultaneous length $\mathrm{O}(N)$ and space $\mathrm{O}(\sqrt{N})$, but any semantic inference-parallel refutation with the same parameters has $\Omega(N)$ configurations with space $\Omega(\sqrt{N})$. We remark that this result makes sense even in the weaker syntactic sequential model, since maximal space $\Omega(\sqrt{N})$ only implies a trivial $\Omega(N)$ cumulative space lower bound.

As already noted, semantic fully parallel resolution is an extremely powerful model, since we can refute any formula with just one (parallel) axiom download step followed by one (semantic) inference step, but if we limit the available space then the usefulness of parallelism is restricted. In fact, the speed-up from parallelism is proportional to the space.

▶ **Observation 17.** *Let π be a semantic parallel resolution refutation of a formula F in length L, maximal clause space s, and cumulative clause space c. Then there is a semantic sequential refutation of F in length Ls, maximal clause space s, and cumulative clause space cs.*

Proof. Each parallel axiom download or inference adds at most s new clauses, therefore we can simulate it by s sequential axiom downloads or inferences respectively. ◀

Using Observation 17 we can transfer the trade-offs above from inference-parallel to fully parallel semantic resolution by sacrificing a factor s.

▶ **Lemma 18.** *Let π be a syntactic sequential resolution refutation of a formula F in length L, maximal space s, and cumulative space c, and let $\ell \in \mathbb{N}^+$ be a positive integer. Then there is a semantic parallel resolution refutation of F in length $3\lceil L/\ell \rceil$, maximal space $s + \lceil \ell/2 \rceil$, and cumulative space $3\lceil c/\ell \rceil + L$.*

Proof sketch. Analogously to the proof of Lemma 3, we divide π into L/ℓ intervals of ℓ steps each. We reorder derivation steps within every interval so that we do all axiom downloads first, inferences next, and removals at the end of the interval. We then collapse each sequence into one axiom download, one inference, and one removal step. ◀

Let us finally just observe that although proving strong lower bounds for the fully parallel versions of resolution looks like a formidable challenge, which we leave as future work, we can obtain a simple separation between semantic and syntactic fully parallel resolution.

▶ **Proposition 19.** *Every syntactic, fully parallel resolution refutation of a minimally unsatisfiable CNF formula in space $s \leq N$ requires length $N/s + \log s - 2$.*

Proof Sketch. Since the inference rule is binary, the number of useful clauses in the second-to-last-last configuration, namely those used to infer contradiction, is at most 2. Analogously, the number of useful clauses in the ith last configuration is at most 2^i. Hence, in the last $\log s$ steps we see at most $2s$ useful clauses in total. Since we need to see each axiom at least once, we still need at least $N/s - 2$ more steps. ◀

In particular, any syntactic refutation requires length $\log N$, and a refutation in this length requires space $\Omega(N)$. This is in contrast to semantic refutations, which have proofs in length 2, and no space lower bound other than the trivial N/L.

By way of example, consider a (plain) pebbling formula on a path graph of length N. A syntactic refutation in length $\log N$ requires space $\Omega(N)$, while there exists semantic refutation in length $\log N$ and space $2N/\log N + \mathrm{O}(1)$: just download the axioms corresponding to $2N/\log N$ consecutive vertices at a time and infer one new clause.

While this is technically a separation, it is also very brittle. For any integer k, it is possible to find a syntactic proof in length $(1+1/k)\log N$ and space $2kN/\log N + \mathrm{O}(1)$. First, download the axioms corresponding to evenly spaced vertices at distance $\log N/k$. Then for $\frac{2}{k}\log N$ steps download the clauses corresponding to the previous and next vertex and do a parallel inference step. Another inference step leaves us with a path of length $N/k \log N$, which we can trivially refute in length $\log N - \log k - \log\log N$ and space $N/k \log N$.

4 Concluding Remarks

In this paper, we study space complexity with a focus not on peak memory usage but on aggregated memory consumption over the whole computation. We consider two computational models, namely pebble games on DAGs and the resolution proof system in proof complexity.

For black-white pebbling, which is a model of nondeterministic computation, we prove optimal cumulative space lower bounds and also time-space trade-offs where in order to achieve optimal time the space needs to be large not only at a single point in time but throughout essentially the whole computation. We do so by studying the concepts of *depth-robustness* and *dispersion* of graphs, drawing on and extending work in [2, 3, 4] and other papers, and proving that different graph families of interest possess these properties.

In the context of proof complexity we are not aware of the cumulative space measure having been studied before, and so our first contribution here is to give a suitable formal definition, and also to consider different, more or less parallel, versions of the resolution proof system in which it makes sense to study cumulative space. We then use, and slightly extend, the reductions between pebbling and resolution in [9, 10] to transfer our lower bounds and trade-off results for pebbling also to resolution.

Since, to the best of our knowledge, ours is the first paper to study cumulative space both for black-white pebbling and for proof complexity, it is perhaps not so surprising that there is a wealth of open problems that this paper does not resolve. Below, we briefly discuss some possible directions for future research.

One set of questions on which we make progress but which we do not answer completely concern the relation between maximal space and cumulative space. For sequential black-white pebblings of n-vertex DAGs we prove an optimal $\Omega\big(n^2\big)$ cumulative space lower bound for a particular family of DAGs, but for graphs that can be pebbled in maximal space $\mathrm{O}(\log n)$ we only obtain a $\Omega\big(n^2/\log n\big)$ cumulative space lower bound and for graphs pebblable in space $\mathrm{O}(1)$ the best cumulative bound we can get is $\Omega\big(n^{3/2}\big)$. Could it be the case that there are graphs that can be pebbled in maximal space $\mathrm{O}(1)$ but nevertheless require cumulative

space $\Omega(n^2)$? Or do strong enough cumulative space lower bounds by necessity imply also nontrivial maximal space lower bounds?

It has been shown for parallel black pebbling that extremal depth-robustness is both necessary and sufficient for a graph to have high cumulative space complexity. We prove that for black-white pebbling predecessor-robustness is sufficient to imply high cumulative space, but leave open whether this condition is necessary or not.

For standard time-space trade-offs in sequential pebbling, it was shown in [37] that bit-reversal DAGs have a black pebbling trade-off of the form $t = \Theta(n^2/s)$ whereas for black-white pebbling the trade-off is a slightly weaker $t = \Theta(n^2/s^2)$. It was conjectured in [37] that there are other permutation graphs for which the black-white pebbling trade-off could also be shown to be an optimal $t = \Theta(n^2/s)$. One natural candidate class of graphs to consider in this context are graphs obtained from random permutations, and this is the original reason why we were interested to study them in this paper. So far we were only able to obtain trade-offs with the same parameters as for bit-reversal DAGs, but it is an interesting question whether our tools could be sharpened to prove even stronger trade-offs results for random permutation graphs.

Turning to our proof complexity results, they can be seen to be yet another contribution to the sequence of papers [39, 43, 9, 40, 10] obtaining space bounds and time-space trade-offs in proof complexity by instead studying pebble games and reductions between pebblings of DAGs and resolution refutations of so-called pebbling formulas defined in terms of these DAGs. While these connections have turned out to be very fruitful, it would also be interesting to go beyond pebbling formulas and explore whether cumulative space results could be obtained for, e.g., Tseitin formulas on long and narrow rectangular grids as studied in [6, 7] or for other formulas.

One motivation behind our models of parallel resolution was the connection to parallel SAT solving, but our models do not take into account practical limitations such as the number of computing nodes or the communication between nodes. Could there be natural ways to incorporate such limitations, and could this also provide a better understanding of parallel resolution?

Another, somewhat related, question is whether formulas possessing strong cumulative space lower bounds are hard also in practice for (sequential or parallel) SAT solvers. Just maximal space lower bounds do not seem to be sufficient to imply practical hardness, as shown, e.g., in the fairly extensive empirical experiments on pebbling formulas in [35], but perhaps cumulative space could be a more relevant concept in this context.

Finally, it can be noted that our study of cumulative space in proof complexity as initiated in this paper is limited to the resolution proof system. This is mostly because resolution is the proof system where space complexity is best understood, and where the toolbox for studying these questions is most well developed. However, different concepts of maximal space and time-space trade-offs have been studied also for other proof systems such as polynomial calculus [1, 7, 13, 26, 27] and cutting planes [21, 29, 31, 34], and it would be interesting to extend the study of cumulative space to these proof systems.

Acknowledgements. The authors wish to thank Yuval Filmus for making us aware of each other's existence and thus providing the stimulus for the joint work that led up to this paper. Different subsets of the authors are grateful to Ilario Bonacina, with whom we had stimulating discussions during various stages of this project and who, in particular, provided valuable insights on dispersion, and to Adam Schill Collberg and Jan Elffers for helpful discussions on random permutation graphs.

References

1. Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Space complexity in propositional calculus. *SIAM Journal on Computing*, 31(4):1184–1211, 2002. Preliminary version in *STOC '00*.
2. Joël Alwen and Jeremiah Blocki. Efficiently computing data-independent memory-hard functions. In *Proceedings of the 36th Annual International Cryptology Conference (CRYPTO '16)*, pages 241–271, August 2016.
3. Joël Alwen and Vladimir Serbinenko. High parallel complexity graphs and memory-hard functions. In *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC '15)*, pages 595–603, June 2015.
4. Joël Alwen, Jeremiah Blocki, and Krzysztof Pietrzak. Depth-robust graphs and their cumulative memory complexity. Technical Report 2016/875, Cryptology ePrint Archive, September 2016.
5. Roberto J. Bayardo Jr. and Robert Schrag. Using CSP look-back techniques to solve real-world SAT instances. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI '97)*, pages 203–208, July 1997.
6. Paul Beame, Chris Beck, and Russell Impagliazzo. Time-space tradeoffs in resolution: Superpolynomial lower bounds for superlinear space. *SIAM Journal on Computing*, 45(4):1612–1645, August 2016. Preliminary version in *STOC '12*.
7. Chris Beck, Jakob Nordström, and Bangsheng Tang. Some trade-off results for polynomial calculus. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC '13)*, pages 813–822, May 2013.
8. Eli Ben-Sasson and Nicola Galesi. Space complexity of random formulae in resolution. *Random Structures and Algorithms*, 23(1):92–109, August 2003. Preliminary version in *CCC '01*.
9. Eli Ben-Sasson and Jakob Nordström. Short proofs may be spacious: An optimal separation of space and length in resolution. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS '08)*, pages 709–718, October 2008.
10. Eli Ben-Sasson and Jakob Nordström. Understanding space in proof complexity: Separations and trade-offs via substitutions. In *Proceedings of the 2nd Symposium on Innovations in Computer Science (ICS '11)*, pages 401–416, January 2011.
11. Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *Journal of the ACM*, 48(2):149–169, March 2001. Preliminary version in *STOC '99*.
12. Ilario Bonacina. Total space in resolution is at least width squared. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP '16)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 56:1–56:13, July 2016.
13. Ilario Bonacina and Nicola Galesi. A framework for space complexity in algebraic proof systems. *Journal of the ACM*, 62(3):23:1–23:20, June 2015. Preliminary version in *ITCS '13*.
14. Ilario Bonacina, Nicola Galesi, and Neil Thapen. Total space in resolution. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS '14)*, pages 641–650, October 2014.
15. María Luisa Bonet, Juan Luis Esteban, Nicola Galesi, and Jan Johannsen. On the relative complexity of resolution refinements and cutting planes proof systems. *SIAM Journal on Computing*, 30(5):1462–1484, 2000. Preliminary version in *FOCS '98*.
16. Siu Man Chan and Aaron Potechin. Tight bounds for monotone switching networks via Fourier analysis. *Theory of Computing*, 10:389–419, October 2014. Preliminary version in *STOC '12*.

17 Ashok K. Chandra. Efficient compilation of linear recursive programs. In *Proceedings of the 14th Annual Symposium on Switching and Automata Theory (SWAT '73)*, pages 16–25, 1973.

18 Vašek Chvátal and Endre Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, October 1988.

19 Stephen A. Cook and Robert Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, March 1979.

20 Stephen A. Cook and Ravi Sethi. Storage requirements for deterministic polynomial time recognizable languages. *Journal of Computer and System Sciences*, 13(1):25–37, 1976. Preliminary version in *STOC '74*.

21 Susanna F. de Rezende, Jakob Nordström, and Marc Vinyals. How limited interaction hinders real communication (and what it means for proof and circuit complexity). In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science (FOCS '16)*, pages 295–304, October 2016.

22 Cynthia Dwork, Moni Naor, and Hoeteck Wee. Pebbling and proofs of work. In *Proceedings of the 25th Annual International Cryptology Conference (CRYPTO '05)*, volume 3621 of *Lecture Notes in Computer Science*, pages 37–54. Springer, August 2005.

23 Patrick W. Dymond and Martin Tompa. Speedups of deterministic machines by synchronous parallel machines. *Journal of Computer and System Sciences*, 30(2):149–161, April 1985. Preliminary version in *STOC '83*.

24 Paul Erdős, Ronald L. Graham, and Endre Szemerédi. On sparse graphs with dense long paths. Technical report, Stanford University, 1975.

25 Juan Luis Esteban and Jacobo Torán. Space bounds for resolution. *Information and Computation*, 171(1):84–97, 2001. Preliminary versions of these results appeared in *STACS '99* and *CSL '99*.

26 Yuval Filmus, Massimo Lauria, Mladen Mikša, Jakob Nordström, and Marc Vinyals. Towards an understanding of polynomial calculus: New separations and lower bounds (Extended abstract). In *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP '13)*, volume 7965 of *Lecture Notes in Computer Science*, pages 437–448. Springer, July 2013.

27 Yuval Filmus, Massimo Lauria, Jakob Nordström, Noga Ron-Zewi, and Neil Thapen. Space complexity in polynomial calculus. *SIAM Journal on Computing*, 44(4):1119–1153, August 2015. Preliminary version in *CCC '12*.

28 Yuval Filmus, Toniann Pitassi, Robert Robere, and Stephen A Cook. Average case lower bounds for monotone switching networks. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS '13)*, pages 598–607, November 2013.

29 Nicola Galesi, Pavel Pudlák, and Neil Thapen. The space complexity of cutting planes refutations. In *Proceedings of the 30th Annual Computational Complexity Conference (CCC '15)*, volume 33 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 433–447, June 2015.

30 John R. Gilbert and Robert Endre Tarjan. Variations of a pebble game on graphs. Technical Report STAN-CS-78-661, Stanford University, 1978. URL: `http://infolab.stanford.edu/TR/CS-TR-78-661.html`.

31 Mika Göös and Toniann Pitassi. Communication lower bounds via critical block sensitivity. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC '14)*, pages 847–856, May 2014.

32 Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39(2-3):297–308, August 1985.

33 John Hopcroft, Wolfgang Paul, and Leslie Valiant. On time versus space. *Journal of the ACM*, 24(2):332–337, April 1977. Preliminary version in *FOCS '75*.

34 Trinh Huynh and Jakob Nordström. On the virtue of succinct proofs: Amplifying communication complexity hardness to time-space trade-offs in proof complexity (Extended abstract). In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC '12)*, pages 233–248, May 2012.

35 Matti Järvisalo, Arie Matsliah, Jakob Nordström, and Stanislav Živný. Relating proof complexity measures and practical hardness of SAT. In *Proceedings of the 18th International Conference on Principles and Practice of Constraint Programming (CP '12)*, volume 7514 of *Lecture Notes in Computer Science*, pages 316–331. Springer, October 2012.

36 George Katsirelos, Ashish Sabharwal, Horst Samulowitz, and Laurent Simon. Resolution and parallelizability: Barriers to the efficient parallelization of SAT solvers. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI '13)*, July 2013.

37 Thomas Lengauer and Robert Endre Tarjan. Asymptotically tight bounds on time-space trade-offs in a pebble game. *Journal of the ACM*, 29(4):1087–1130, October 1982. Preliminary version in *STOC '79*.

38 João P. Marques-Silva and Karem A. Sakallah. GRASP: A search algorithm for propositional satisfiability. *IEEE Transactions on Computers*, 48(5):506–521, May 1999. Preliminary version in *ICCAD '96*.

39 Jakob Nordström. Narrow proofs may be spacious: Separating space and width in resolution. *SIAM Journal on Computing*, 39(1):59–121, May 2009. Preliminary version in *STOC '06*.

40 Jakob Nordström. On the relative strength of pebbling and resolution. *ACM Transactions on Computational Logic*, 13(2):16:1–16:43, April 2012. Preliminary version in *CCC '10*.

41 Jakob Nordström. Pebble games, proof complexity and time-space trade-offs. *Logical Methods in Computer Science*, 9:15:1–15:63, September 2013.

42 Jakob Nordström. New wine into old wineskins: A survey of some pebbling classics with supplemental results. Manuscript in preparation. To appear in *Foundations and Trends in Theoretical Computer Science*. Current draft version available at `http://www.csc.kth.se/~jakobn/research/`, 2017.

43 Jakob Nordström and Johan Håstad. Towards an optimal separation of space and length in resolution. *Theory of Computing*, 9:471–557, May 2013. Preliminary version in *STOC '08*.

44 Michael S. Paterson and Carl E. Hewitt. Comparative schematology. In *Record of the Project MAC Conference on Concurrent Systems and Parallel Computation*, pages 119–127, 1970.

45 Wolfgang J. Paul and Rüdiger Reischuk. On alternation II. A graph theoretic approach to determinism versus non-determinism. *Acta Informatica*, 14(4):391–403, 1980. Preliminary version in *GITCS '79*.

46 Nicholas Pippenger. Pebbling. Technical Report RC8258, IBM Watson Research Center, 1980. in Proceedings of the 5th IBM Symposium on Mathematical Foundations of Computer Science, Japan.

47 Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, March 1999. Preliminary version in *FOCS '97*.

48 John E. Savage. *Models of Computation: Exploring the Power of Computing*. Addison-Wesley, 1998. URL: `http://www.modelsofcomputation.org`.

49 Georg Schnitger. On depth-reduction and grates. In *Proceedings of the 24th Annual IEEE Symposium on Foundations of Computer Science (FOCS '83)*, pages 323–328, November 1983.

50 Ravi Sethi. Complete register allocation problems. *SIAM Journal on Computing*, 4(3):226–248, September 1975.

51 Sowmitri Swamy and John E. Savage. Space-time trade-offs on the FFT-algorithm. Technical Report CS-31, Brown University, 1977.

52 Alasdair Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, January 1987.

53 Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In *Proceedings of the 6th International Symposium on Mathematical Foundations of Computer Science (MFCS '77)*, pages 162–176, September 1977.

A Hierarchy Theorem for Interactive Proofs of Proximity

Tom Gur[*1] **and Ron D. Rothblum**[†2]

1 **Weizmann Institute, Rehovot, Israel**
tom.gur@weizmann.ac.il
2 **MIT, Cambridge, USA**
rothblum@gmail.com

Abstract

The number of rounds, or round complexity, used in an interactive protocol is a fundamental resource. In this work we consider the significance of round complexity in the context of *Interactive Proofs of Proximity* (IPPs). Roughly speaking, IPPs are interactive proofs in which the verifier runs in sublinear time and is only required to reject inputs that are far from the language.

Our main result is a round hierarchy theorem for IPPs, showing that the power of IPPs grows with the number of rounds. More specifically, we show that there exists a gap function $g(r) = \Theta(r^2)$ such that for every constant $r \geq 1$ there exists a language that (1) has a $g(r)$-round IPP with verification time $t = t(n, r)$ but (2) does not have an r-round IPP with verification time t (or even verification time $t' = \mathsf{poly}(t)$).

In fact, we prove a stronger result by exhibiting a *single* language $\mathcal{L}$ such that, for every constant $r \geq 1$, there is an $O(r^2)$-round IPP for $\mathcal{L}$ with $t = n^{O(1/r)}$ verification time, whereas the verifier in *any* r-round IPP for $\mathcal{L}$ must run in time at least t^{100}. Moreover, we show an IPP for $\mathcal{L}$ with a poly-logarithmic number of rounds and only poly-logarithmic verification time, yielding a sub-exponential separation between the power of constant-round IPPs versus general (unbounded round) IPPs.

From our hierarchy theorem we also derive implications to standard interactive proofs (in which the verifier can run in polynomial time). Specifically, we show that the round reduction technique of Babai and Moran (JCSS, 1988) is (almost) optimal among all blackbox transformations, and we show a connection to the algebrization framework of Aaronson and Wigderson (TOCT, 2009).

1998 ACM Subject Classification F.1.3 [Computation by Abstract Devices] Complexity Measures and Classes

Keywords and phrases Complexity Theory, Property Testing, Interactive Proofs

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.39

1 Introduction

Interactive Proofs, introduced by Goldwasser at el. [38] (and in their public-coin form, by Babai and Moran [8]), are protocols in which a computationally unbounded prover tries to convince a verifier that an input x belongs to a language $\mathcal{L}$. A recent line of work, initiated by Ergün, Kumar and Rubinfeld [19] and more recently by Rothblum, Vadhan and

* Tom Gur is supported by the ISF grant number 671/13 and Irit Dinur's ERC grant number 239985; part of this research was conducted while visiting Columbia University, New York.
† Ron Rothblum is partially supported by NSF MACS - CNS-1413920 and by SIMONS Investigator award Agreement Dated 6-5-12.


© Tom Gur and Ron D. Rothblum;
licensed under Creative Commons License CC-BY

Wigderson [58], considers a variant of interactive proofs in which the verifier is required to run in *sublinear* time. Since the verifier does not have enough time to even read its entire input, we cannot expect it to reject every false statement. Rather, following the property testing literature [59, 27] (see also [26]), we relax the soundness condition and only require that the verifier reject inputs that are *far* from the language (no matter what cheating strategy the prover uses). Since the verifier is only assured that the input is close to the language, such interactive proofs are called *interactive proofs of proximity* (IPPs). Indeed, IPPs may be thought of as the property testing analogue of interactive proofs.

From an information theoretic perspective, the key parameters of an IPP are its *query complexity*, *communication complexity* and *round complexity*. The *query complexity* is the number of bits of the input string that the verifier reads. The *communication complexity* is the number of bits exchanged between the prover and the verifier, and the *round complexity* is the number of rounds of interaction. We think of all of these parameters as being *sublinear* in the input length. Additional computational parameters that we aim to minimize are the verifier's running time (which should also be sublinear) and the prover's running time (which, ideally, should be proportional to the complexity of deciding the language).

In this work we focus on the round complexity of IPPs, and on the relation between the number of rounds and the other parameters. Specifically, we ask the following question:

Does the power of Interactive Proofs of Proximity grow with the number of rounds?

Understanding the round complexity of protocols is a central problem in the theory of computation (most notably in complexity theory and cryptography). Some of the main motivations for reducing round complexity are considerations such as network latency, the need to stay online or to synchronize messages between the parties, and the overhead involved in sending and receiving messages.

1.1 Our Results

Our main result answers the foregoing question by showing a hierarchy of IPPs: we show that for a gap function $g(r) = \Theta(r^2)$, and for every constant $r \geq 1$, it holds that r-round IPPs can be outperformed by $g(r)$-round IPPs, in the sense that the verifier in the latter system is significantly more efficient. We prove our hierarchy theorem by constructing a *single* explicit language for which the power of IPPs grows with the number of rounds.

▶ **Theorem 1** (Hierarchy theorem, informally stated (see Theorem 16)). *There exists an explicit language $\mathcal{L}$ such that for every constant $r \geq 1$ and for inputs of length n:*

1. *There is an $O(r^2)$-round* IPP *for $\mathcal{L}$ in which the verifier runs in time $t = n^{O(1/r)}$; and*
2. *The verifier in* any *r-round* IPP *for $\mathcal{L}$* must *run in time at least $t' = t^{100}$ (where the constant* 100 *is arbitrary). Furthermore, either the communication complexity or query complexity of the verifier must be at least t'.*

Thus, we obtain a characterization (which is exact, up to the specific polynomial of the gap function g) of the complexity of constant-round IPPs for the language $\mathcal{L}$.

For simplicity, the statement in Theorem 1 is restricted to constant-round protocols. However, the complexity of the IPP protocol in Theorem 1 actually reduces further as the round complexity grows to be super-constant. In particular, we obtain a poly-logarithmic round IPP for $\mathcal{L}$ with poly-logarithmic communication and query complexities, and an $\omega(1)$-round IPP with $n^{o(1)}$ communication and query complexities. Together with the lower bound in Theorem 1, these yield a separation between the power of constant-round IPPs and super-constant round IPPs, and a *sub-exponential* separation with respect to poly-logarithmic round IPPs.

▶ **Theorem 2** (Constant Round versus General IPPs). *There exists a language $\mathcal{L}$ that has a* polylog(n)*-round* IPP *with a* polylog(n) *time verifier and an $\omega(1)$-round* IPP *with $n^{o(1)}$ time verifier, but for every constant $r \geq 1$, the verifier in any r-round* IPP *for $\mathcal{L}$ must run in time at least $n^{\Omega(1/r)}$.*

Prior to this work, only a separation between the power of MAPs (which are *non-interactive* IPPs, i.e., the entire "interaction" consists of a *single* message) and IPPs was known [43].

We remark that Theorems 1 and 2, and their proofs, shed new light also on standard interactive proofs (in which the verifier is given direct access to the input and can run in polynomial time). We proceed to discuss such implications.

Optimality of the Babai-Moran Round Reduction. Following Vadhan [66], we consider *black-box transformations on interactive proofs*, which are transformations that take prover and verifier strategies $(\mathcal{P}, \mathcal{V})$, for an interactive-proof for some language $\mathcal{L}$, and output new strategies $(\mathcal{P}', \mathcal{V}')$, for the same language $\mathcal{L}$, such that new prover and verifier strategies can only make oracle calls to the original strategies $(\mathcal{P}, \mathcal{V})$. More specifically, the new verifier $\mathcal{V}'$ is only allowed to make oracle calls to $\mathcal{V}$ (and in particular does not have direct access to the input) and $\mathcal{P}'$ may make oracle calls to both $\mathcal{V}$ and $\mathcal{P}$.[1]

As pointed out by Vadhan, many (but not all) of the known transformations on interactive proofs from the literature are in fact black-box. We focus on such a transformation, due to Babai and Moran [8], for reducing the number of rounds of interaction in public-coin interactive proofs. Using our hierarchy theorem, we show that the overhead incurred by the round reduction transformation of [8] is close to optimal among all black-box transformations.

Algebrization of Interactive Proofs. As our second application, we show a connection between our hierarchy theorem and the *algebrization* framework of Aaronson and Wigderson [1]. This framework, which is an extension of the *relatization* framework of Baker, Gill, and Solovay [9], is viewed as a barrier to proving complexity-theoretic lower bounds using currently known proof techniques. Loosely speaking, [1] show that almost all known complexity theoretic results "algebrize" (i.e., fall within their framework), whereas making progress on some of our most fundamental questions (such as $\mathcal{P} \neq$ NP) requires non-algebrizing techniques.

Using our hierarchy theorem for IPPs, we show that any proof of the complexity class inclusion $\#\mathcal{P} \subseteq \mathcal{AM}$ (which is widely disbelieved, and in particular implies the collapse of the polynomial hierarchy) must make use of non-algebrizing techniques, and therefore *must* introduce a fundamentally different proof technique. A conceptual connection between our results and interactive proofs in the algebrization framework is further discussed in Section 1.3 and elaborated on in Section 5.

1.2 Technical Overview

Loosely speaking, the language for which we prove the round hierarchy theorem consists of error-correcting encodings of strings $x \in \{0,1\}^k$ whose Hamming weight $\mathsf{wt}(x) \stackrel{\text{def}}{=} \sum_{i \in [k]} x_i$, is divisible by 3 (i.e., $\mathsf{wt}(x) = 0 \pmod 3$).

[1] One could also restrict $\mathcal{P}'$ to make only oracle calls to $\mathcal{P}$ (and not to $\mathcal{V}$ as we do). However, giving $\mathcal{P}'$ more freedom only makes our results stronger (since we rule out the broader class of transformations).

The specific encoding that we use is the low degree extension code $\mathsf{LDE} : \mathbb{F}^k \to \mathbb{F}^n$, over a field $\mathbb{F}$ that is an extension field of $\mathsf{GF}(2)$.[2] Indeed, it is crucial that the characteristic of $\mathbb{F}$ is different than the modulus 3. The parameter k (which specifies the message length) is the same as in the preceding paragraph, where we view $\{0,1\}^k$ as a subset of the message space $\mathbb{F}^k$.

Before proceeding, we note that throughout this work we use the standard convention that codes map messages of length k to codewords of length $n = \mathsf{poly}(k)$. In particular, this will mean that inputs to IPPs, which will typically refer to (possibly corrupt) codewords, have length n, whereas inputs to other types of protocols and sub-routines, may refer to the underlying messages, which have length k.

Recall that the LDE code is parameterized by a finite field $\mathbb{F}$, a subset of the field $H \subseteq \mathbb{F}$ and a dimension m. To encode a message $x \in \{0,1\}^k$, where $k = |H|^m$, we view the message as a function $x : H^m \to \{0,1\}$ (by associating $[k]$ with H^m) and consider the unique individual degree $|H| - 1$ polynomial $P : \mathbb{F}^m \to \mathbb{F}$ that agrees with x on H^m. We denote this polynomial by $P = \mathsf{LDE}_{\mathbb{F},H,m}(x)$. For the time being, the sizes of $|\mathbb{F}|$, $|H|$ and m should all be thought of as at most poly-logarithmic in n. (See Section 2.3 for additional details about the LDE encoding.)

Thus, the language for which we prove our round hierarchy, which we denote by Enc-MOD3, consists of all polynomials $P : \mathbb{F}^m \to \mathbb{F}$ of individual degree $|H| - 1$ that obtain Boolean values in the subcube H^m, such that these Boolean values sum up to 0 (mod 3). That is, all polynomials P such that $P|_{H^m} : H^m \to \{0,1\}$ and $\sum_{z \in H^m} P(z) \equiv 0 \pmod 3$.

We prove our hierarchy theorem by showing that for every constant $r \geq 1$, the language Enc-MOD3 has an $O(r^2)$-round IPP in which the verifier runs in time roughly $n^{O(1/r)}$, and that the verifier in *any* r-round IPP for Enc-MOD3 must run in time at least $n^{\Omega(1/r)}$, where the constant in the Ω-notation can be made arbitrarily larger than the constant in the O-notation. In Section 1.2.1 we give an overview of the upper bound, which is technically more involved, and then, in Section 1.2.2 we give an overview of the lower bound.

1.2.1 Upper Bound

Our goal is to construct an IPP in which the verifier is given oracle access to a function $f : \mathbb{F}^m \to \mathbb{F}$ and needs to verify that f is close to a polynomial of individual degree $|H| - 1$ that obtains only Boolean values in H^m such that their sum modulo 3, over the subcube H^m, is 0. The verifier may interact with the prover for $O(r^2)$ rounds.

As its initial step, our verifier checks that the given input f is close to some low degree polynomial by invoking the low degree test. This test, introduced by Rubinfeld and Sudan [59], ensures that if f is far from every low degree polynomial, then the verifier will reject with high probability. Thus, we can assume that f is close to some low degree polynomial. Moreover, using the self-correction property of polynomials, this means that with a small overhead, we can treat f as though it were itself a low degree polynomial (rather than just being close).[3]

Given this initial step, we can now assume without loss of generality that the function $f : \mathbb{F}^m \to \mathbb{F}$ is in fact a low degree polynomial. However, the verifier still needs to check that $\sum_{z \in H^m} f(z) = 0 \pmod 3$ and that $f|_{H^m} : H^m \to \{0,1\}$. For now though, let us focus

[2] We remark that a similar result could be obtained if we replaced the modulus 3 and the field's characteristic by any two *distinct* and *constant-sized* primes.

[3] Loosely speaking, the self-correction property of polynomials says that if f is guaranteed to be close to a low degree polynomial P, then one can read values from P by only making few queries to f. See Lemma 30 for the precise statement.

on the former task, which is the main step in our proof: checking that $\sum_{z \in H^m} f(z) = 0 \pmod 3$ (and we just assume that $f|_{H^m} : H^m \to \{0,1\}$).

Viewing $f|_{H^m}$ as a string $x \in \{0,1\}^k$, we need to construct an interactive proof in which the verifier uses oracle access to $\mathsf{LDE}(x)$ to verify that $\mathsf{wt}(x) = 0 \pmod 3$ in sublinear time. We refer to this type of proof-system, in which the verifier is given oracle access to an *encoding* of the input and runs in sublinear time, as a *holographic*[4] *interactive proof* (HIP).

More precisely, we say that a language has an HIP, with respect to some error-correcting code C, if it has an interactive proof in which the verifier has oracle access to an encoding under C of the input and verifies membership in the language using few queries to this encoding. The redundant representation of the input often allows the verifier to run in *sub-linear time*. We remark that HIPs play a central role in this work and we discuss them more in Section 1.3.

Thus, our task is now to construct an HIP (with respect to the LDE code) for the language

$$\mathcal{L}_{\mathsf{MOD3}} \stackrel{\text{def}}{=} \left\{ x \in \{0,1\}^k \ : \ \mathsf{wt}(x) = 0 \pmod 3 \right\}.$$

Before describing the construction of an HIP for $\mathcal{L}_{\mathsf{MOD3}}$, it will be instructive to consider as a warm-up, the construction of an HIP for the related language $\mathcal{L}_{\mathsf{MOD2}} = \{x \in \{0,1\}^k : \mathsf{wt}(x) = 0 \pmod 2\}$, where the important distinction is that the modulus 2 is also the characteristic of the field $\mathbb{F}$ under which x is encoded.

In this warmup case, we assume that the verifier is given oracle access to a polynomial $f : \mathbb{F}^m \to \mathbb{F}$ that obtains Boolean values in H^m (i.e. $f|_{H^m} : H^m \to \{0,1\}$), and needs to check that $\sum_{z \in H^m} f(z) = 0$, where the sum is over $\mathsf{GF}(2)$. Importantly, since we assumed that $f|_{H^m}$ is Boolean valued, and that the field $\mathbb{F}$ has characteristic 2, we can instead take the sum over the field $\mathbb{F}$ (rather than taking the integer sum mod 2).

The latter problem, of checking whether the sum of a given input polynomial is 0 over a subcube of its domain (i.e., over H^m), has a well-known interactive proof due to Lund *et al.* [52], which is often referred to as the *sumcheck protocol*. In this protocol the verifier only needs to query the polynomial f at a single point and so it can be viewed as an HIP. Furthermore, there are known variants of the sumcheck protocol that offer a suitable tradeoff between the number of rounds and verifier's complexity, which suffice for our purposes (i.e., an r-round IPP with verification time roughly $n^{1/r}$).

The aforementioned variants of the sumcheck protocol suffice for an upper bound for the warmup case. However, we do not know how to prove a corresponding lower bound, which is the reason that we set the modulus in our construction to be different from the field's characteristic.[5] While the original language $\mathcal{L}_{\mathsf{MOD3}}$ allows us to prove the desired lower bound, unfortunately it makes obtaining an upper bound more challenging. We proceed to the actual problem at hand: constructing an HIP (with respect to the LDE code over a field of characteristic 2) for checking $\mathcal{L}_{\mathsf{MOD3}}$.

Since the modulus and characteristic are different, our task can no longer be expressed as a linear constraint (over $\mathbb{F}$) on the bits of x. Since we do not know how to solve this problem directly using the sumcheck protocol, we turn to more complex interactive proofs from the literature. Specifically, our starting point will be the interactive proof-system of Goldwasser, Kalai and Rothblum [37], which we shall refer to as the GKR protocol.[6]

[4] The terminology of "holographic" interactive proofs originates from the "holographic proofs" of Babai *et al.* [5], which refers to probabilistic proof systems for *encoded* inputs. The notion of HIP, and its relation to other notions, is further discussed in Section 1.3.

[5] We conjecture that for the warmup case (i.e., when the modulus is 2) a lower bound that (roughly) corresponds to the upper bound given by the sumcheck protocol does hold.

[6] In Section A.1 we discuss our reason for basing our protocol on the GKR proof-system, rather than

The GKR Protocol. Goldwasser *et al.* give an interactive proof for any language computable by a logspace-uniform circuit of size S and depth D such that the number of rounds in their protocol is $D \cdot \mathsf{polylog}(S)$, the communication is also $D \cdot \mathsf{polylog}(S)$, and the verifier runs in time $(n + D) \cdot \mathsf{polylog}(S)$. Their protocol is based on algebraic techniques and, in particular, uses ideas originating from the interactive proof and PCP literature (cf., [63]). Our HIP for MOD3 will be based on a variant of their proof system.

Observe that one can check whether a given string x's Hamming weight is divisible by 3 using a highly uniform logarithmic-depth formula.[7] Thus, applying the GKR result gives us an interactive proof for $\mathcal{L}_{\mathsf{MOD3}}$. Most importantly for our purposes, if the GKR verifier is given oracle access to the LDE encoding of the input, then it only needs to check a single (random) element from the encoding (and in particular runs in sublinear time). In other words, the GKR protocol can be thought of as an HIP with *sublinear* time verification.[8] While the GKR protocol does yield an HIP for $\mathcal{L}_{\mathsf{MOD3}}$, its round complexity is poly-logarithmic and therefore too large for our purposes (recall that we are aiming for constant round protocols). The large round complexity is due to the fact that the high-level strategy in the GKR protocol is to process the circuit layer by layer, where the transition between each two consecutive layers uses an interactive protocol, which itself is based on the sumcheck protocol.

Even if we were to use a constant-round variant of the sumcheck protocol for each transition, the GKR protocol still uses $\Omega(D)$ rounds, where D is the depth of the circuit, which in our case is logarithmic and therefore too large. To get around this, we rely on an unpublished observation, due to Kalai and Rothblum [45], which shows that for every constant $r \geq 1$, if the circuit satisfies an extreme (and somewhat unnatural) uniformity condition[9], then $\log(n)/r$ layers can be processed at once, using r rounds of interaction and roughly $n^{1/r}$ communication. Thus, overall, a logarithmic depth circuit can processed in $O(r^2)$ rounds. Using this observation, [45] obtain *constant-round* interactive-proofs for all languages in NC^1 that satisfy the aforementioned uniformity condition.[10]

In our actual construction we do not use the [45] protocol directly (even though the language $\mathcal{L}_{\mathsf{MOD3}}$ satisfies the desired uniformity), but rather give a special purpose protocol tailored for $\mathcal{L}_{\mathsf{MOD3}}$ (which is inspired by their techniques). Doing so allows us to avoid stating their somewhat cumbersome uniformity condition and to introduce other simplifications (due to the simple and regular structure of the formula for $\mathcal{L}_{\mathsf{MOD3}}$). We proceed to describe this HIP.

---

other general purpose interactive proof-systems from the literature.

[7] E.g., consider the $\log(k)$-depth full binary tree with the input bits at its leaves, in which each internal vertex computes the sum modulo 3 of its two children, where each such modulo 3 sum can be computed by a simple constant size gadget composed of AND, OR and NOT gates.

[8] Note that obtaining an interactive-proof for $\mathcal{L}_{\mathsf{MOD3}}$ with a *linear-time* verifier is trivial, since the verifier can decide membership by itself in linear-time. The key benefit that we get from using the GKR protocol is that it allows for *sublinear* time verification given access to an encoded input.

[9] Loosely speaking, the uniformity condition requires that it be possible to compute *low degree extensions* of gate indicator functions that refer to gates of fan-in $t = n^{O(1/r)}$. That is, we view the formula as a depth r circuit consisting of gates of fan-in $t = n^{O(1/r)}$ (by grouping together every $\log(n)/r$ consecutive layers). For each of these r layers, and every type of fan-in t gate $g : \{0,1\}^t \to \{0,1\}$ that appears in that layer, we consider a gate indicator function I_g that given as input indices of $t + 1$ wires, outputs 1 if the first wire is the result of an application of g to the other t wires. The [45] uniformity requirement is that it be possible to efficiently compute the *low degree extension* of I_g.

[10] Recall that the class NC^i consists of languages computable by polynomial-size $O\left((\log n)^i\right)$-depth circuits with fan-in 2. We emphasize that the [45] result gives *constant-round* protocols only for NC^1 circuits (that are sufficiently uniform), whereas the GKR result gives protocol with a *poly-logarithmic* round complexity for all (logspace uniform) languages in $\mathsf{NC} = \cup_{k \in \mathbb{N}} \mathsf{NC}^k$. (Furthermore, the GKR protocol for NC has poly-logarithmic communication complexity whereas the [45] protocol has $n^{1/O(1)}$ communication.)

A Holographic Interactive Proof for $\mathcal{L}_{\mathsf{MOD3}}$. Recall that we are given oracle access to a polynomial $X : \mathbb{F}^m \to \mathbb{F}$ promised to be the low-degree extension of a *Boolean* assignment $x \in \{0,1\}^k$, and our goal is to construct an $O(r^2)$-round HIP for verifying whether $x \in \mathcal{L}_{\mathsf{MOD3}}$. Also recall that we have fixed the parameters of the LDE code, including a field $\mathbb{F}$, a subset $H \subseteq \mathbb{F}$, and a dimension m such that $|H^m| = k$. However, for now we think of the sizes of these parameters as being $|H| = k^{1/r}$, $m = r$, and $|\mathbb{F}| = \mathsf{poly}(|H|, m)$, rather than $|H|$ being poly-logarithmic in k.[11]

For a given input polynomial $X : \mathbb{F}^m \to \mathbb{F}$ (of individual degree $|H| - 1$), we define a sequence of polynomials $V_0, \ldots, V_r$, where each $V_i : \mathbb{F}^i \to \mathbb{F}$ has individual degree $|H|-1$ (note that these polynomials have gradually increasing domains). The polynomial $V_r : \mathbb{F}^r \to \mathbb{F}$ is defined as $V_r \equiv X$. The polynomials $V_1, \ldots, V_{r-1}$ are each defined to be the (unique) individual degree $|H| - 1$ polynomial that satisfies the following recursive relation:

$$\forall i \in [r],\ \forall h \in H^{i-1}, \quad V_{i-1}(h) = \sum_{\alpha \in H} V_i(h, \alpha) \pmod 3, \tag{1}$$

where the arithmetic is over the integers (modulo 3). Indeed, $V_0 \in \mathbb{F}$ is defined as a single field element $V_0 = \sum_{\alpha \in H} V_1(\alpha) \pmod 3$. Note that we identify the integers $\{0,1,2\}$ with three distinct elements in $\mathbb{F}$. Indeed, each of the V_i polynomials takes values in the set $\{0,1,2\} \subseteq \mathbb{F}$ over the subcube H^i.

Taking the [37, 45] view, each polynomial $V_i : \mathbb{F}^i \to \mathbb{F}$ can be thought of as the low degree extension of the i^{th}-layer (counting from the output layer) in a depth r formula of fan-in $k^{1/r}$ for $\mathcal{L}_{\mathsf{MOD3}}$ such that each gate computes the sum modulo 3 of its $k^{1/r}$ children. In particular,

$$V_0 = \sum_{\alpha \in H} V_1(\alpha) = \cdots = \sum_{h \in H^i} V_i(h) = \cdots = \sum_{h \in H^r} V_r(h) = \mathsf{wt}(x) \pmod 3.$$

Our main step is an interactive protocol that reduces a claim about an (arbitrary) single point in the polynomial V_{i-1} to a claim about a single (random) point in V_i. By applying this interactive reduction r times, we can reduce the initial claim $V_0 = 0$ to a claim about a single point in V_r, which we can explicitly check (since we have oracle access to $V_r \equiv X$). Each interactive reduction will take $O(r)$ rounds so overall we get an HIP for $\mathcal{L}_{\mathsf{MOD3}}$ with $O(r^2)$ rounds.

Towards showing such an interactive reduction protocol, we would like to express Equation (1), which is a modular equation over the integers, as a low degree relation over the field $\mathbb{F}$. Let $t \stackrel{\text{def}}{=} |H| = k^{1/r}$, and let $\xi_1, \ldots, \xi_t$ be the enumeration of all elements in H. Define the polynomial $\widetilde{\mathsf{MOD3}} : \mathbb{F}^t \to \mathbb{F}$ as the (unique) individual degree two polynomial such that for every $z \in \{0,1,2\}^t$, it holds that $\widetilde{\mathsf{MOD3}}(z) = \sum_{j \in [t]} z_i \pmod 3$, where the tilde in the notation is meant to remind us that $\widetilde{\mathsf{MOD3}}$ is not the modulo 3 summation function but rather its low degree extension over $\mathbb{F}$. Equation (1) can now be re-stated as:

$$\forall i \in [r],\ \forall h \in H^{i-1}, \quad V_{i-1}(h) = \widetilde{\mathsf{MOD3}}\Big(V_i(h, \xi_1), \ldots, V_i(h, \xi_t)\Big) \tag{2}$$

(where we use the fact that the V_i polynomials take values in $\{0,1,2\}$ over H^i.)

[11] We remark that setting $|H| = k^{1/r}$ is actually problematic for us since it induces a dependence between the language $\mathsf{Enc\text{-}MOD3}$ and the desired round complexity r. Nevertheless, it does yield a weaker hierarchy theorem in which we use a different language for each value of r. At the end of Section 1.2.1 we discuss how we overcome this difficulty.

Observe that Equation (2) is a polynomial relation between V_{i-1} and V_i that holds for inputs in H^{i-1}. We would like to obtain a similar relation for general inputs (i.e., in $\mathbb{F}^{i-1}$). To do so, we observe that, for every $z \in \mathbb{F}^{i-1}$, we can express $V_{i-1}(z)$ as an $\mathbb{F}$-linear combination of the values $\{V_{i-1}(h)\}_{h\in H^{i-1}}$ (this follows directly from the fact that the low degree extension is a *linear* code). We denote the coefficients in this linear combination by $\{\beta_z(h)\}_{h\in H^{i-1}}$ (these coefficients arise from Lagrange interpolation, but we ignore the specifics for this overview). Combining this observation together with Equation (2) we obtain:

$$\begin{aligned}\forall i \in [r],\ \forall z \in \mathbb{F}^{i-1}, \quad V_{i-1}(z) &= \sum_{h\in H^{i-1}} \beta_z(h)\cdot V_{i-1}(h) \\ &= \sum_{h\in H^{i-1}} \beta_z(h)\cdot \widetilde{\mathsf{MOD3}}\Big(V_i(h,\xi_1),\ldots,V_i(h,\xi_t)\Big). \end{aligned} \tag{3}$$

Using Equation (3) we will describe an interactive reduction from a claim about V_{i-1} to a claim about V_i. Suppose that our interactive reduction starts with a claim that $V_{i-1}(z_{i-1}) = \nu_{i-1}$ for some $z_{i-1} \in \mathbb{F}^{i-1}$ and $\nu_{i-1} \in \mathbb{F}$. By Equation (3) this translates into the claim:

$$\nu_{i-1} = \sum_{h\in H^{i-1}} \beta_{z_{i-1}}(h)\cdot \widetilde{\mathsf{MOD3}}\Big(V_i(h,\xi_1),\ldots,V_i(h,\xi_t)\Big). \tag{4}$$

We now observe that $Q_i(w) \stackrel{\text{def}}{=} \beta_{z_{i-1}}(w)\cdot \widetilde{\mathsf{MOD3}}\Big(V_i(w,\xi_1),\ldots,V_i(w,\xi_t)\Big)$ is a low degree polynomial over $\mathbb{F}$ (since $\beta_{z_{i-1}}$, $\widetilde{\mathsf{MOD3}}$, and V_i have low degree). Thus, the claim in Equation (4) refers to the sum of a low degree polynomial over a subcube, which is precisely the problem that the sumcheck protocol solves.

It seems that we are done, except that a problem arises. In the sumcheck protocol the verifier is given oracle access to the polynomial whose sum over a subcube we wish to check. Although the polynomial Q_i on which we wish to run the sumcheck protocol is well-defined, our verifier does not have oracle access to it. Therefore it is not immediately clear how we can hope to run the sumcheck protocol with respect to Q_i.

We resolve this problem by noting that the sumcheck protocol can be used in an *input-oblivious* manner. In this variant, the verifier does not need to have oracle access to Q_i, but rather than accepting or rejecting, the verifier outputs a claim of the form $Q_i(w_{i-1}) = \gamma_{i-1}$, for some point $w_{i-1} \in \mathbb{F}^{i-1}$ and value $\gamma_{i-1} \in \mathbb{F}$. Completeness means that if the original claim is true (i.e., $\sum_{h\in H^{i-1}} Q_i(h) = \nu_{i-1}$), then the verifier always outputs (w_{i-1}, γ_{i-1}) such that $Q_i(w_{i-1}) = \gamma_{i-1}$, and soundness means that if the original claim is false (i.e., $\sum_{h\in H^{i-1}} Q_i(h) \neq \nu_{i-1}$), then for any cheating prover strategy, with high probability $Q_i(w_{i-1}) \neq \gamma_{i-1}$ (or the verifier rejects during the interaction). We stress that in this variant the verifier makes no queries to Q_i.[12] As for the number of rounds, recall that in the sumcheck protocol in each iteration one of the variables is "stripped" from the summation, which leads to a total of $i - 1 \leq r$ rounds.

Having run the input-oblivious variant of the sumcheck protocool, our verifier is now left with the claim $Q_i(w_{i-1}) = \gamma_{i-1}$. However, to obtain our interactive reduction, we still need to reduce the foregoing claim to a claim about a (single) point in the polynomial V_i. To do so, the first idea that comes to mind is to have the prover provide the values

[12] To see that this variant is possible, observe that in the classical sumcheck protocol [52], the verifier only queries the polynomial at a single point and (at the end of the interaction) checks that it is equal to a particular value.

$\mu_j = V_i(w_{i-1}, \xi_j)$, for every $j \in [t]$. Given these values, the verifier can explicitly check that indeed $\gamma_{i-1} = \beta_{z_{i-1}}(w_{i-1}) \cdot \widetilde{\mathsf{MOD3}}(\mu_1, \ldots, \mu_t)$.[13] If the prover indeed sent the correct values, then this last check assures us that indeed $Q_i(w_{i-1}) = \gamma_{i-1}$. However, since we cannot assume that the prover sent the correct values, we are left with t claim of the form $V_i(w_{i-1}, \xi_j) = \mu_j$, which the verifier needs to check.

Notice that we have actually reduced a single claim about V_{i-1} to t claims about V_i. This still falls short of our goal which was to reduce to only a *single* claim about V_i. (Indeed, we cannot afford to increase the number of claims by a t factor in each iteration, since this would yield a protocol with complexity $t^r = k$, which is trivial).

The final observation is that the points $\{(w_{i-1}, \alpha)\}_{\alpha \in H}$ lie on the (axis parallel) line $(w_{i-1}, *)$. Note that the restriction of a low degree polynomial to an axis parallel line is a low degree (univariate) polynomial. Thus, we will have the prover specify the entire polynomial $P_i : \mathbb{F} \to \mathbb{F}$ defined as $P_i(\alpha) = V_i(w_{i-1}, \alpha)$, for every $\alpha \in \mathbb{F}$. The verifier checks that $\gamma_{i-1} = \beta_{z_{i-1}}(w_{i-1}) \cdot \widetilde{\mathsf{MOD3}}(P_i(\xi_1), \ldots, P_i(\xi_t))$. The point is that now if the prover supplies an incorrect values for some $P_i(\alpha)$ (i.e., $P_i(\alpha) \neq V_i(w_{i-1}, \alpha)$), since both P_i and $V_i(w_{i-1}, *)$ are low degree polynomials, for most $\rho \in \mathbb{F}$ it holds that $P_i(\rho) \neq V_i(w, \rho)$. Thus, the verifier chooses at random $\rho_i \in \mathbb{F}$ and sets the claim for the next iteration to be $V_i(z_i) = \nu_i$, where $z_i = (w_{i-1}, \rho_i)$ and $\nu_i = P_i(\rho_i)$.[14]

To summarize, our HIP for $\mathcal{L}_{\mathsf{MOD3}}$ works in r phases. In the i^{th} phase we reduce a claim of the form $V_{i-1}(z_{i-1}) = \nu_{i-1}$, for some point $z_{i-1} \in \mathbb{F}^{i-1}$ and value $\nu_{i-1} \in \mathbb{F}$, into a claim $V_i(z_i) = \nu_i$, for $z_i \in \mathbb{F}^i$ and $\nu_i \in \mathbb{F}$ (which are generated during the interactive reduction). In particular, the first iteration begins with the claim $V_0 = 0$ (i.e., z_0 is the empty string and $\nu_0 = 0$), which corresponds to the claim that $x \in \mathcal{L}_{\mathsf{MOD3}}$ (i.e., $\mathsf{wt}(x) = 0 \pmod 3$). Thus, the i^{th} phase in our HIP begins with the claim $V_{i-1}(z_{i-1}) = \nu_{i-1}$. In the i^{th} phase, first the prover and verifier engage in the sumcheck protocol that arises from Equation (4). This yields the claim $Q_i(w_{i-1}) = \gamma_{i-1}$, for a point $w_{i-1} \in \mathbb{F}^{i-1}$ and value $\gamma_{i-1} \in \mathbb{F}$ (generated by the sumcheck protocol). Since the verifier has no access to Q_i, it asks the prover to send the polynomial $P_i : \mathbb{F} \to \mathbb{F}$ defined as $P_i(\alpha) = V_i(w_{i-1}, \alpha)$. The verifier checks that the values of this polynomial are consistent with the claim $Q_i(w_{i-1}) = \gamma_{i-1}$, and then selects a random point $\rho_i \in \mathbb{F}$. The claim for the following phase is that $V_i(z_i) = \nu_i$, where $z_i = (w_{i-1}, \rho_i)$ and $\nu_i = P_i(\rho_i)$. After r such phases we are left with the claim $V_r(z_r) = \nu_r$, for $z_r \in \mathbb{F}^r$ and $\nu_r \in \mathbb{F}$, which the verifier can explicitly check (since it has oracle access to $V_r \equiv X$).

The total number of rounds per interactive reduction is $O(r)$, and the communication complexity is roughly $\mathsf{poly}(t, r) = \mathsf{poly}(r, k^{1/r})$. Since we invoke r such reductions, overall we obtain an HIP for $\mathcal{L}_{\mathsf{MOD3}}$ with round complexity $O(r^2)$ and communication complexity $\mathsf{poly}(r, k^{1/r})$.

Obtaining an HIP over a Small Field. The approach outlined above yields an r^2-round HIP for $\mathcal{L}_{\mathsf{MOD3}}$, with respect to the code $\mathsf{LDE}_{\mathbb{F},H,m}$, in which the field size $|\mathbb{F}|$ is quite large (i.e., $|\mathbb{F}| \geq k^{1/r}$) and in particular depends on the value of r. Unfortunately, when we transform this HIP into an IPP for the language Enc-$\mathsf{MOD3}$, the dependence of the field size on r in the HIP introduces a dependence of the language Enc-$\mathsf{MOD3} \stackrel{\text{def}}{=} \{C(x) \,:\, x \in \{0,1\}^k \text{ with } \mathsf{wt}(x) = 0 \pmod 3\}$ on r. This dependence results in a weaker hierarchy theorem, in which we use a

[13] Note that both $\beta_{z_{i-1}}$ and $\widetilde{\mathsf{MOD3}}$ are *explicit* functions that the verifier can compute. Moreover they can even be computed *efficiently* using standard techniques, see the technical sections for details.

[14] We remark that this final step is actually very reminiscent of an individual round of the sumcheck protocol.

different language for each value of r . Our goal however is to obtain a *single* language, for which we can show an r-round IPP for every value of r (with a corresponding lower bound, which will be discussed in Section 1.2.2).

To this end we show a general reduction that transforms any HIP over a large field $\mathbb{F}$ into an HIP over a much smaller field $\mathbb{F}'$, as long as $\mathbb{F}$ is an extension field of $\mathbb{F}'$. We do so by showing that any $\mathbb{F}$-linear claim regarding the input (e.g., a claim about a single point in the $\mathsf{LDE}_{\mathbb{F},H,m}$ encoding) can be broken down (coordinate-wise) into d claims that are $\mathbb{F}'$-linear, where $d = \log(|\mathbb{F}|/|\mathbb{F}'|)$ is the degree of the field extension (i.e., $(\mathbb{F}')^d$ is isomorphic to $\mathbb{F}$). We can then easily verify each one of these $\mathbb{F}'$-linear claims using the sumcheck protocol over the smaller field $\mathbb{F}'$.[15]

We remark that the ability to switch fields when using (holographic) interactive proofs seems like a useful tool, and we believe that it will be useful in other contexts as well.

Checking Booleanity. In the above analysis we assumed for simplicity that the input $x = f|_{H^m}$ is Boolean valued. In order to actually check this, we follow an idea of Kalai and Raz [46] (which was used in the context of constructing interactive PCPs). We observe that the polynomial $f : \mathbb{F}^m \to \mathbb{F}$ is Boolean valued in a subcube H^m if and only if the (slightly higher degree) polynomial $g : \mathbb{F}^m \to \mathbb{F}$, defined as $g(z) = f(z) \cdot (1 - f(z))$ is identically 0 in H^m. The latter problem (of checking whether a polynomial vanishes on a particular subcube) can be solved via a relatively simple reduction to the sumcheck protocol, that has been used in the construction of PCPs.[16] We note that we crucially use fact that the reduction from f to g is local (i.e., the value of g at a point depends on the value of f at $O(1)$ points), and therefore can be used in our setting.

1.2.2 Lower Bound

We need to show a lower bound on the complexity of r-round IPPs for our language $\mathsf{Enc\text{-}MOD3} = \{C(x) \ : \ x \in \{0,1\}^k \text{ with } \mathsf{wt}(x) = 0 \pmod 3\}$, where $C : \mathbb{F}^k \to \mathbb{F}^n$ is the low degree extension code. Our lower bound will strongly use the fact that any $\mathbb{F}$-linear code (and in particular the low degree extension code that we use), for a field $\mathbb{F}$ of characteristic 2, is also a GF(2)-linear code.

Our lower bound relies on a connection between IPPs and low-depth circuits, which was discovered by Rothblum, Vadhan and Wigderson [58]. Following their approach, in Section 4.3 we show that to prove an IPP lower bound for Enc-MOD3, it suffices to construct two distributions D_0 and D_1 over n-bit strings such that:

1. D_0 is distributed over the support of Enc-MOD3 (with high probability);
2. D_1 is far from Enc-MOD3 (with high probability); and
3. Every sufficiently small DNF formula cannot distinguish between inputs from D_0 and D_1 (with more than, say, 0.1 advantage).

[15] To obtain the desired *computational* efficiency for the latter task, we actually use the [45] protocol, which introduces an additional $O(r^2)$ rounds. We believe this use is an overkill, and we hope to replace it with a more elementary argument in a following revision.

[16] In a nutshell, to check whether $g|_{H^m} \equiv 0$ we consider the restriction of g to the domain H^m and take the low degree extension $\hat{g}$ of that partial function. We observe that g is identically 0 in H^m if and only if $\hat{g}$ is identically 0 in $\mathbb{F}^m$. Thus, it suffices to check whether for a random point $z \in \mathbb{F}^m$, which the verifier chooses, it holds that $\hat{g}(z) = 0$. The linearity of the LDE code now means that this check can be solved by invoking the sumcheck protocol. See Section 3.4 for details.

The two distributions that we consider are D_0 and D_1 such that D_b is uniform over the set $\{C(x) \ : \ x \in \{0,1\}^k \text{ and } \mathsf{wt}(x) = b \pmod 3\}$. Note that D_0 is the uniform distribution over Enc-MOD3, and so satisfies requirement (1), whereas the fact that D_1 satisfies requirement (2) follows from the distance of the code C. To show that the third requirement holds, consider a DNF ϕ that distinguishes between D_0 and D_1. We show that the size of ϕ must be large. Consider the distributions D'_0 and D'_1 over k-bit strings defined as

$$D'_b = \{x \in \{0,1\}^k \ : \ \mathsf{wt}(x) = b \pmod 3\}.$$

We can easily construct from ϕ a circuit ϕ' that distinguishes between D'_0 and D'_1: the circuit ϕ' first computes the encoding $C(x)$ of its input $x \in \{0,1\}^k$, and then applies the DNF ϕ to the result. Using the fact that C is linear over $\mathsf{GF}(2)$, it follows that ϕ' is a DNF of parities (i.e., a depth-3 formula with an OR gate at the top layer, AND gates at the middle layer, and XOR gates at the bottom layer). Now, we can apply the Razborov-Smolensky [59] lower bound, which shows that any small $\mathsf{AC}_0[2]$ circuit (i.e., circuits of constant-depth circuits with AND, OR, and PARITY gates of unbounded fan-in), and in particular a DNF of parities, cannot even approximate the summation modulo 3 function (i.e., distinguish between D'_0 and D'_1).

1.3 Holographic Interactive Proofs

The proof of our hierarchy theorem utilizes a special type of interactive proofs, which we call *holographic interactive proofs*. A holographic interactive proof (HIP) is an interactive proof in which, instead of getting its input x explicitly, the verifier is given *oracle* access to $C(x)$, an error-corrected encoding of the input x, for a bounded number of queries. Hence, HIPs may be thought of as interactive proofs for promise problems of the form $(\Pi_{\mathsf{YES}}, \Pi_{\mathsf{NO}})$ with $\Pi_{\mathsf{YES}} = \{C(x) : x \in \mathcal{L}\}$ and $\Pi_{\mathsf{NO}} = \{C(x) : x \notin \mathcal{L}\}$.

The notion of HIP was used, either implicitly or explicitly as a technical tool that underlies many probabilistic proof systems (e.g., [52, 6, 5, 46, 37, 47, 58, 43, 48, 56, 28]).[17] These works demonstrate that, by using the redundant encoding of the input, we can often achieve sublinear verification time. (As a matter of fact, in most of these works, it suffices for the verifier to read just a *single* point in the encoding.) We remark that throughout this work (as well as in most previous works[18]), the specific code that is used is the *low-degree extension code* (LDE).

Some of the techniques that were outlined in Section 1.2.1, can be viewed as generic transformations on HIPs (with respect to the LDE code), and we present them as such in the technical parts of this work. These techniques include the ability to switch fields, or check Booleanity, and the connection to IPPs. We wish to highlight the conceptual importance of HIPs, and advocate a continued systematic study of these proof systems.

We also remark that HIPs with respect to the LDE code are closely related to interactive proofs in the algebrization framework [1]. In both models the verifier is given oracle access to a low degree polynomial and may interact with the prover to decide on some property of the "message" or "oracle" encoded within the polynomial. See Section 5 for further discussion of this connection.

[17] The first explicit use is in [5].

[18] A notable exception is the work of Meir [53], which is based on general tensor codes. We remark that using Meir's techniques it may be possible to extend our results to other tensor codes. We leave exploring this possibility to future work.

1.4 Related Works

In this section, we discuss several lines of works that are related to our work.

Interactive Proofs of Proximity. The notion of interactive proofs of proximity (IPP) was first considered by Ergün, Kumar and Rubinfeld [19]. Its study was re-initiated by Rothblum, Vadhan and Wigderson [58], who showed that every language computable by a low-depth circuit has an IPP with a sublinear time verifier. IPPs were further studied by [31, 28] who showed more efficient IPPs for certain restricted complexity classes. Other works have focusing on variants such as non-interactive (MA) proofs of proximity [43, 20, 30] and interactive *arguments* of proximity [49]. Proofs of proximity have also found applications to property testing and related models [33, 34, 21].

Hierarchy Theorems for Standard Interactive Proofs. Aiello, Goldwasser and Håstad [2] showed a round hierarchy theorem in a relativized world (i.e., with respect to an oracle). However, the later results of [52, 60], which are based on non-relativizing techniques, demonstrate that relativization is not an actual barrier, especially in the context of interactive proofs.[19] We note that although they are technically quite different, both our lower bound and the lower bound of [2] are based on circuit lower bounds for low depth circuits.

Goldreich, Vadhan and Wigderson [36] showed a *conditional* round hierarchy result for standard interactive proofs, based on the assumption that co-SAT does not have a 1-round $\mathcal{AM}$ proof-system with complexity $2^{o(n)}$.[20] We emphasize that the result of [36] is based on an unproven and arguably strong (yet believable) assumption, whereas our result is unconditional.

We also note that for *computationally sound* proofs, also known as arguments, under reasonable cryptographic assumptions there are extremely efficient 2-round protocols [50] and even 1-round protocols [48]. In particular, these results show that the power of arguments does not scale with additional rounds (since a fixed constant number of rounds suffice). A similar statement holds for arguments of proximity that are the computationally sound variant of IPPs (see [58, 49]).

Interactive PCPs. Holographic interactive proofs (HIPs) are closely related to the notion of *interactive* PCPs, introduced by Kalai and Raz [46]. Roughly speaking, interactive-PCPs are encodings of NP-witnesses that, like PCPs can be verified using few queries, but here the verification procedure may use interaction with an unbounded (and untrusted) prover. Thus, using our terminology, an interactive PCP can be thought of as an HIP for checking the NP witness relation.

Arthur-Merlin Query Complexity. Every IPP for a language $\mathcal{L}$ can be viewed as a protocol, for a promise problem related to $\mathcal{L}$, in the Arthur Merlin query complexity model, previously studied by Raz *et al.* [54]. This model, similarly to IPPs, considers a sub-linear time verifier, that is given oracle access to an input and may interact with an (untrusted) prover. Indeed, one may view IPPs as Arthur Merlin query complexity protocols which focus on promise problems in which the goal of the verifier is to distinguish between inputs having a certain property from those that are *far* from having the property.

---

[19] Indeed, Fortnow and Sipser [22] show that the proof of IP = PSPACE cannot be relativized (in fact, IP does not even contain coNP relative to a random oracle [14]). In fact, the algebrization framework of Aaronson and Wigderson [1] was proposed precisely to address this issue. Connections between our results and algebrization are further discussed in Section 5.

[20] Related assumptions have recently been studied also by Carmosino *et al.* [10] and Williams [69].

Thus, our main result directly yields a round hierarchy theorem (for a promise problem) in the *Arthur-Merlin Query Complexity* model and a sub-exponential separation between the complexity of constant-round vs. general (i.e., unbounded round) Arthur-Merlin Query Complexity protocols.

Interactive Proofs in Other Models. Interactive proof systems were studied also in the communication complexity setting (e.g., [7, 51, 61, 41, 40]). Here Alice and Bob may interact with an untrusted Merlin, who sees both of their inputs. We remark that showing any non-trivial explicit lower bound in the $\mathcal{AM}$ variant of this model, much less a hierarchy of separations, is a notorious open problem.

A recent line of works has studied interactive proofs in the data streaming model (e.g., [12, 16, 17, 42, 11, 65, 18]). Most relevant is a result of Chakrabarti *et al.* [13], who show a hierarchy theorem for the first four levels in the model of *online interactive proofs* (with exponential separations between these four levels).

Universal Locally Verifiable Codes. In a recent work, Goldreich and Gur [29] introduced the notion of *universal locally verifiable codes* (universal-LVC), which is closely related to holographic interactive proofs. A universal-LVC $C : \{0,1\}^k \to \{0,1\}^n$ for a family of functions $\mathcal{F} = \{f_i : \{0,1\}^k \to \{0,1\}\}_{i\in[M]}$ is a code such that for every $i \in [M]$, membership in the subcode $\{C(x) : f_i(x) = 1\}$ can be verified locally given an explicit access to a short (sublinear length) proof; put differently, for every $i \in [M]$ there exists a 1-message IPP for the property $\{C(x) : f_i(x) = 1\}$, with sublinear communication and query complexity.

1.5 Organization

In Section 2 we define IPPs and introduce some notations and definitions that we use throughout this work. In Section 3 we define holographic interactive proofs (HIPs) and prove some general results on them. In Section 4, using some of the results of Section 3, we prove the hierarchy theorem. Lastly, in Section 5 we discuss the implications to classical complexity theory.

Some of the discussion and proofs are deferred to the appendix. In Appendix A.2 we discuss an alternative language for the round hierarchy theorem and our choice of basing our protocol on GKR rather than, say a recent protocol of Reingold *et al.* [56]. Appendices B to D contain some standard proofs that are included for completeness.

2 Preliminaries

We begin with some standard notations:

- We denote the relative distance, over alphabet Σ, between two strings $x \in \Sigma^n$ and $y \in \Sigma^n$ by $\Delta(x,y) \stackrel{\text{def}}{=} \frac{|\{x_i \neq y_i \,:\, i\in[n]\}|}{n}$. If $\Delta(x,y) \leq \varepsilon$, we say that x is ε-close to y, and otherwise we say that x is ε-far from y. Similarly, we denote the relative distance of x from a non-empty set $S \subseteq \Sigma^n$ by $\Delta(x,S) \stackrel{\text{def}}{=} \min_{y\in S} \Delta(x,y)$. If $\Delta(x,S) \leq \varepsilon$, we say that x is ε-close to S, and otherwise we say that x is ε-far from S.
- We denote the projection of $x \in \Sigma^n$ to a subset of coordinates $I \subseteq [n]$ by $x|_I$ and, for $i \in [n]$, write $x_i = x|_{\{I\}}$ to denote the projection to a singleton.

An additional notation that we will use is that if $S = (S_k)_{k\in\mathbb{N}}$ and $T = (T_k)_{k\in\mathbb{N}}$ are ensembles of sets, we denote by $S \subseteq T$ the fact that $S_k \subseteq T_k$ for every $k \in \mathbb{N}$.

Integrality. Throughout this work, for simplicity of notation, we use the convention that all (relevant) integer parameters that are stated as real numbers are implicitly rounded to the closest integer.

2.1 Interactive Proofs of Proximity

A language is an ensemble $\mathcal{L} = (\mathcal{L}_n)_{n\in\mathbb{N}}$, where $\mathcal{L}_n \subseteq (\Sigma_n)^n$ for every $n \in \mathbb{N}$ and where $\Sigma = (\Sigma_n)_{n\in\mathbb{N}}$ is the alphabet.

▶ **Definition 3** (Interactive Proofs of Proximity (IPP))**.** Let $\Sigma = (\Sigma_n)_{n\in\mathbb{N}}$ be an alphabet ensemble. An r-round *interactive proof of proximity*, with respect to proximity parameter $\varepsilon > 0$, (in short, ε-IPP) for the language $\mathcal{L}$ is an interactive protocol between a prover $\mathcal{P}$, which gets *free* access to ε and to an input $x \in \Sigma^n$, and a verifier $\mathcal{V}$, which gets free access only to ε and n, as well as *oracle* access to x. At the end of the protocol, the following conditions are satisfied:

- **Completeness**: If $x \in \mathcal{L}$, then, when $\mathcal{V}$ interacts with $\mathcal{P}$, with probability 2/3 it accepts.
- **Soundness**: If x is ε-far from $\mathcal{L}$, then for every prover strategy $\mathcal{P}^*$, when $\mathcal{V}$ interacts with $\mathcal{P}^*$, with probability 2/3 it rejects.

If the completeness condition in Definition 3 holds with probability 1, then we say that the IPP has perfect completeness. A public-coin IPP is an IPP in which every message from the verifier to the prover consists only of fresh random coin tosses.

An IPP is said to have query complexity $q : \mathbb{N} \times [0,1] \to \mathbb{N}$ if for every $n \in \mathbb{N}$, $\varepsilon > 0$, $x \in \{0,1\}^n$, and any prover strategy $\mathcal{P}^*$, the verifier makes at most $q(n,\varepsilon)$ queries to x when interacting with $\mathcal{P}^*$. The IPP is said to have communication complexity $c : \mathbb{N} \times [0,1] \to \mathbb{N}$ if for every $n \in \mathbb{N}$, $\varepsilon > 0$, and $x \in \mathcal{L}_n$ the communication between $\mathcal{V}$ and $\mathcal{P}$ consists of at most $c(n,\varepsilon)$ bits.

2.2 Constructible Error Correcting Codes and Finite Fields

An error correcting code over an alphabet Σ is an injective function $C : \Sigma^k \to \Sigma^n$. The code C is said to have relative distance δ if for any $x \neq x' \in \Sigma^k$ it holds that $\Delta(x, x') \geq \delta$.

Throughout this work we deal with (uniform) polynomial-time algorithms, and so we will need (families of) codes that are efficiently computable. Formally, for a parameter $n = n(k) \geq 1$ that is called the blocklength, and ensemble of alphabets $\Sigma = (\Sigma_k)_{k\in\mathbb{N}}$, we define a constructible error correcting code over Σ as an ensemble $C = \left(C_k : \Sigma_k^k \to \Sigma_k^n\right)_{k\in\mathbb{N}}$ of error correcting codes, such that the function $f(x) = C_{|x|}(x)$ is computable by a polynomial-time Turing machine (in particular this implies that $n = \mathsf{poly}(k, \log(\Sigma))$). An ensemble of error correcting codes $C = (C_k)_{k\in\mathbb{N}}$ is said to have relative distance δ if for all sufficiently large k, each code C_k in the ensemble has relative distance δ.

Throughout this work, we mostly consider codes defined over finite fields (i.e., the alphabets Σ_k are all finite fields). Such codes are called linear if they are linear functions over the field.

Finite Fields and Polynomials. Many of our algorithms and interactive proofs deal with finite fields. We consider ensembles of finite fields $\mathbb{F} = (\mathbb{F}_k)_{k\in\mathbb{N}}$, where $|\mathbb{F}_k|$ and say that such ensembles are constructible if the field operations can be done in $\mathsf{poly}\log(|\mathbb{F}_k|)$ time. Namely, there exist a Turing machine that given as input k and an appropriate number of elements in $\mathbb{F}_k$ (represented as strings of length $O(\log(|\mathbb{F}_k|))$ bits) can compute the field operations (i.e., addition, subtraction, multiplication, inversion, and sampling random elements) in $\mathsf{polylog}(|\mathbb{F}_k|)$ time.

The following fact shows that there exist constructible finite fields of characteristic 2.

▶ **Fact 4.** *For every time-constructible function $f = f(k) \geq 1$, there exists a* constructible *field ensemble $\mathbb{F} = (\mathbb{F}_k)_{k\in\mathbb{N}}$ such that $|\mathbb{F}| = O(f)$ and $\mathbb{F}_k$ has characteristic 2 (i.e., is an extension field of $\mathsf{GF}(2)$) for every $k \in \mathbb{N}$.*

For details see [24, Appendix G.3] and references therein. We will also use the well-known Schwartz-Zippel Lemma.

▶ **Lemma 5** (Schwartz-Zippel Lemma)**.** *Let $P : \mathbb{F}^m \to \mathbb{F}$ be a non-zero polynomial of total degree d over the field $\mathbb{F}$. Then,*

$$\Pr_{x\in_R\mathbb{F}^m}[P(x) = 0] \leq \frac{d}{|\mathbb{F}|}.$$

2.3 Low-Degree Extension

Let $\mathbb{F} = (\mathbb{F}_k)_{k\in\mathbb{N}}$ be an ensemble of fields, and let $H = (H_k)_{k\in\mathbb{N}} \subseteq \mathbb{F}$ (the notation $H \subseteq \mathbb{F}$ means that $H_k \subseteq \mathbb{F}_k$, for every $k \in \mathbb{N}$). Let $m = m(k) \geq 1$ be a parameter, which we often call the dimension.

A basic fact is that for every function $f : H^m \to \mathbb{F}$ there exists a *unique* function $\tilde{f} : \mathbb{F}^m \to \mathbb{F}$ such that $\tilde{f}$ is a polynomial with individual degree $|H| - 1$ that agrees with f on H^m. Moreover, there exists an individual degree $|H| - 1$ polynomial $\beta : \mathbb{F}^m \times \mathbb{F}^m \to \mathbb{F}$ such that for every function $f : H^m \to \mathbb{F}$ it holds that

$$\tilde{f}(z) = \sum_{x\in H^m} \beta(x, z) \cdot f(x).$$

The function $\tilde{f}$ is called the low degree extension of $\mathbb{F}$ (with respect to the field $\mathbb{F}$, subset H and dimension m).

The following two propositions show that the low degree extension encoding can be computed efficiently.

▶ **Proposition 6.** *Let $\mathbb{F} = (\mathbb{F}_k)_{k\in\mathbb{N}}$ be a constructible field ensemble, let $H = (H_k)_{k\in\mathbb{N}} \subseteq \mathbb{F}$ be an ensemble of subsets and let $m = m(k)$ be the dimension.*

There exists a Turing machine that on input k runs in time $\mathsf{poly}(|H|, m, \log|\mathbb{F}|)$ and space $O(\log(|\mathbb{F}|) + \log(m))$, and outputs the polynomial $\beta : \mathbb{F}^m \times \mathbb{F}^m \to \mathbb{F}$ defined above, represented as an arithmetic circuit over $\mathbb{F}$.

Moreover, the arithmetic circuit β can be evaluated in time $\mathsf{poly}(|H|, m, \log(|\mathbb{F}|))$ and space $O(\log(|\mathbb{F}|) + \log(m))$. Namely, there exists a Turing machine with the above time and space bounds that given an input pair $(x, z) \in \mathbb{F}^m \times \mathbb{F}^m$ outputs $\beta(x, z)$.

See, e.g., [57, Proposition 3.2.1] for a proof of Proposition 6.

▶ **Proposition 7.** *Let $\mathbb{F} = (\mathbb{F}_k)_{k\in\mathbb{N}}$ be a constructible field ensemble, let $H = (H_k)_{k\in\mathbb{N}} \subseteq \mathbb{F}$ be an ensemble of subsets and let $m = m(k)$ be the dimension.*

Let $\phi : H^m \to \mathbb{F}$ and suppose that ϕ can be evaluated by a Turing Machine in time t and space s. Then, there exists a Turing machine that, given as an input a point $z \in \mathbb{F}^m$, runs in time $|H|^m \cdot (\mathsf{poly}(|H|, m, \log(|\mathbb{F}|)) + O(t))$ and space $O(m \cdot \log(|H|) + s + \log(|\mathbb{F}|))$ and outputs the value $\hat{\phi}(z)$ where $\hat{\phi}$ is the unique low degree extension of ϕ (with respect to $H, \mathbb{F}, m$).

Proof. The Turing machine computes

$$\hat{\phi}(z) = \sum_{x\in H^m} \beta(x, z) \cdot \phi(x)$$

by generating and evaluating β as in Proposition 6. ◀

Low Degree Extension as an Error-Correcting Code. The low degree extension can also be viewed as an error-correcting code in the following way. Suppose that H and m are such that $|H|^m = k$. Then, we can associate a string $x \in \mathbb{F}^k$ with a function $x : H^m \to \mathbb{F}$ by identifying H^m with $[k]$ in some canonical way.

We define the low degree extension of a string x as $\mathsf{LDE}_{\mathbb{F},H,m}(x) = \tilde{x}$. That is, the function $\mathsf{LDE}_{\mathbb{F},H,m}$ is given as input the string $x \in \mathbb{F}^k$, views it as a function $x : H^m \to \mathbb{F}$ and outputs its low degree extension $\tilde{x}$. By Proposition 7 the code $\mathsf{LDE}_{\mathbb{F},H,m}$ is constructible, and by the Schwartz-Zippel Lemma (Lemma 5), the code $\mathsf{LDE}_{\mathbb{F},H,m}$ has relative distance $1 - \frac{m \cdot |H|}{|\mathbb{F}|}$.

3 Holographic Interactive Proofs

In this section we define *holographic interactive proofs* and show several transformations and generic results (which will be used in Section 4 for the proof of the hierarchy theorem). In Section 3.1 we give a formal definition and some basic facts. Having read Section 3.1, the reader may freely skip the rest of Section 3 and proceed directly to Section 4, which is the main technical section, and return to read the results of Sections 3.2 to 3.4 when they are used in Section 4.

Sections 3.2 to 3.4 focus on HIPs with respect to the low degree extension encoding. In Section 3.2 we show that such HIPs imply interactive proofs of *proximity* (for a related language). In Section 3.3 we show that one can switch the field under which the HIPs input is encoded (at a moderate cost) to any other field *that shares the same characteristic*. Finally, in Section 3.4 we show that HIPs can efficiently verify that the input (which can presumably be an arbitrary vector over the field) is actually Boolean valued (i.e., in $\{0,1\}^k$).

3.1 Definition and Basic Facts

A *holographic interactive proof* is similar to a standard interactive proof, except that rather than getting the input explicitly, the verifier gets oracle access to an encoding of the input (via an error correcting code). Using this redundant representation, we could potentially hope to have protocols in which the verifier runs in *sublinear* time and, in particular, does not even read its entire input. This hope is indeed materialized in several protocols from the literature (e.g., [52, 37, 56]).

As a matter of fact, it turns out that for some codes (specifically the low degree extension), reading just a single point p from the encoded input suffices for the verifier.[21] Thus, we restrict our attention to such protocols. Furthermore, in order to facilitate composition, rather than having the verifier actually read the (encoded) input at the point p, the verifier outputs a claim about the point (i.e., it outputs p together with a symbol that it would have expected to see, had it actually queried the (encoded) input at p).

Formally, holographic interactive proofs are parametrized by a (constructible) error correcting code C, under which the input is encoded, and are defined as follows.

▶ **Definition 8** (Holographic Interactive Proofs (HIP))**.** Let $\Sigma = (\Sigma_k)_{k\in\mathbb{N}}$ and $\Lambda = (\Lambda_k)_{k\in\mathbb{N}}$ be alphabet ensembles such that $\Lambda \subseteq \Sigma$. Let $\mathcal{L} \subseteq \Lambda$, and let $C : \Sigma^k \to \Sigma^n$ be a constructible error correcting code.

[21] For the low degree extension this can be shown to hold generically. The high level idea is to consider a low degree curve passing through all the points that the verifier wishes to read. The prover specifies the values for all the points on the curve and the verifier checks the provided answer on a random point on the curve. Soundness follows from the fact that composing a low-degree curve with a low-degree polynomial results in a low degree univariate polynomial. See, e.g., [46, Section 6] for details.

An r-round public-coin *holographic interactive proof (*HIP*)* for the language $\mathcal{L}$, with respect to the code C, is an interactive protocol between a prover $\mathcal{P}$, which gets as input $x \in \Sigma^k$, and a verifier $\mathcal{V}$, which gets as input only k. At the end of the protocol either the verifier rejects or it outputs a coordinate $i \in [n]$ and a symbol $\sigma \in \Sigma$ such that:

- **Completeness**: If $x \in \mathcal{L}$, then, when $\mathcal{V}$ interacts with $\mathcal{P}$, with probability 1 it outputs (i, σ) such that $C(x)|_i = \sigma$.
- **Soundness**: If $x \notin \mathcal{L}$, then for every prover strategy $\mathcal{P}^*$, when $\mathcal{V}$ interacts with $\mathcal{P}^*$, with probability $1 - \varepsilon$ either $\mathcal{V}$ rejects or it outputs (i, σ) such that $C(x)|_i \neq \sigma$, where $\varepsilon = \varepsilon(k) \in [0, 1]$ is called the *soundness error*.

In this work, all the holographic proofs that we consider are with respect to the low degree extension code (using a variety of different parameters), which was defined in Section 2.3 above.

▶ Remark (Different Alphabets for the Language and the Code). Typically, when using HIPs the alphabet Λ over which the language is defined will be the same as the alphabet Σ over which the code is defined. Still, in some cases it will be convenient for us to present HIPs that only work for particular sub-alphabets of the code (e.g., when the input is binary but the code is more naturally defined over some large alphabet) and so we give this more flexible definition.

Our definition of HIPs tries to capture many of the known interactive proof-systems in the literature, while being flexible and easy to compose. Indeed, the fact that HIPs can be transformed into standard interactive proofs which is immediate, is captured by the following proposition.

▶ **Proposition 9.** *Let $\Sigma = (\Sigma_k)_{k\in\mathbb{N}}$ and $\Lambda = (\Lambda_k)_{k\in\mathbb{N}}$ be alphabets such that $\Lambda \subseteq \Sigma$. Let $\mathcal{L}$ be a language over the alphabet Λ and let $C : \Sigma^k \to \Sigma^n$ be a constructible error correcting code.*

Any HIP *for $\mathcal{L}$ can be converted into a standard interactive proof with only a* poly(n) *additive overhead to the verifier's running time (and all other parameters remain unchanged). Moreover, the precise overhead is equal to the time that it takes to compute the i^{th} character of $C(x)$, given $x \in \Lambda^k$ and the index $i \in [n]$.*

Proof. The prover and verifier run the HIP. If the HIP verifier rejects, then we immediately reject. Otherwise, the HIP verifier outputs a pair $(i, \sigma) \in [n] \times \Sigma$ with the associated claim $C(x)|_i = \sigma$. We can now check this claim directly by computing $C(x)|_i$ and comparing with σ. ◀

The Sumcheck Protocol (as an HIP). We will make extensive use of the classical sumcheck protocol of Lund *et al.* [52]. Recall that the sumcheck protocol is an interactive proof for verifying that the sum, over a subcube, of a low degree polynomial is zero. Our protocol differs slightly from the "textbook" sumcheck protocol in two ways:

1. The verifier does not actually read any points from the input polynomial. Rather, at the end of the protocol it outputs a claim about a single point of the polynomial (i.e., the protocol is an HIP).
2. Following other works in the literature, our protocol allows a trade-off between the number of rounds and the communication complexity (rather than having the number of rounds correspond exactly to the dimension of the polynomial).

▶ **Lemma 10** (Sumcheck as an HIP). *Let $\mathbb{F}$ be a constructible field ensemble and let $H \subseteq \mathbb{F}$ be an ensemble of subsets of $\mathbb{F}$. Let $m = m(k)$ be an ensemble of integers such that $m = \log_{|H|}(k)$.*

Let $\mathcal{L} = \cup_{k\in\mathbb{N}}\mathcal{L}_k$, where $\mathcal{L}_k = \{x \in \mathbb{F}^k : \sum_{i\in[k]} x_i = 0\}$ and where the summation is over the field $\mathbb{F}$. Then, for every $r \in [m]$, there exists an r-round (public-coin) HIP for $\mathcal{L}$, with respect to the code $\mathsf{LDE}_{\mathbb{F},H,m}$, with soundness error $\frac{m\cdot|H|}{|\mathbb{F}|}$ and communication complexity $|H|^{\lceil m/r\rceil}\cdot r\cdot\log|\mathbb{F}|$. The verifier runs in time $|H|^{\lceil m/r\rceil}\cdot r\cdot\mathsf{polylog}(|\mathbb{F}|)$ and the prover runs in time $\mathsf{poly}(|\mathbb{F}|^m, r)$.

The proof of Lemma 10, which is standard, is included for completeness in Appendix C.

3.2 From HIP to IPPs

Proposition 9 above, shows that an HIP can be easily transformed into a standard interactive proof. We now show that HIPs, with respect to the low degree extension encoding, can be easily transformed into highly efficient (and in particular sublinear) *interactive proof of proximity* (IPP) for a related language. More specifically, we transform an HIP for the language $\mathcal{L}$ with respect to the $\mathsf{LDE}_{\mathbb{F},H,m}$ code, into an IPP for the language $\mathsf{LDE}_{\mathbb{F},H,m}(\mathcal{L}) \stackrel{\text{def}}{=} \{\mathsf{LDE}_{\mathbb{F},H,m}(x) : x \in \mathcal{L}\}$.[22]

▶ **Lemma 11.** *Let $\mathbb{F} = (\mathbb{F}_k)_{k\in\mathbb{N}}$ be an ensemble of finite fields, let $H = (H_k)_{k\in\mathbb{N}}$ be an ensemble of subsets (i.e. $H \subseteq \mathbb{F}$) and let $m = m(k)$ be such that $|H|^m = k$.*

Suppose that the language $\mathcal{L}$ has an r-round HIP, with respect to the code $\mathsf{LDE}_{\mathbb{F},H,m}$, with communication complexity c. Then, the language $\mathsf{LDE}_{\mathbb{F},H,m}(\mathcal{L})$ has an r-round ε-IPP with query complexity $O(|H|\cdot m\cdot 1/\varepsilon)$ and communication complexity c.

The key observations that we use to prove Proposition 11 are that (1) the IPP verifier can first check that its input is close to a low degree polynomial using low degree test. If the test passes, then, using the self-correctability of polynomials, the IPP verifier can emulate access to the encoded input of the HIP. Given these two observations the proof of Proposition 11 is standard and so we defer it to Appendix B.

3.3 Field Switching

In this subsection we show that HIPs can evaluate points in a LDE over an *extension* field of the base field under which the input is actually encoded. This fact is used in the proof Lemma 17 and allows us to first construct an HIP over a large field, and later convert it into an HIP over the smaller field.

The key observation for our field switching, is that verifying a linear claim involving the LDE over an extension field $\mathbb{K}/\mathbb{F}$ can be reduced to verifying several linear claims over the base field $\mathbb{F}$. Each of these linear claims can be verified via a sumcheck protocol (in fact, it suffices to verify a random linear combination of these claims), and so an HIP can emulate access to the LDE over the extension field $\mathbb{K}$ by making queries to the LDE over field $\mathbb{F}$. We proceed to the formal statement and proof.

Let $\mathbb{F} = (\mathbb{F}_k)_{k\in\mathbb{N}}$ and $\mathbb{K} = (\mathbb{K}_k)_{k\in\mathbb{N}}$ be constructible field ensembles such that $\mathbb{K}$ is a degree $s = s(k) \le \log(k)$ field extension of $\mathbb{F}$ (i.e., $\mathbb{K}_k \cong \mathbb{F}_k^{s(k)}$, for every $k \in \mathbb{N}$). Let $H = (H_k)_{k\in\mathbb{N}} \subseteq \mathbb{F}$ and $G = (G_k)_{k\in\mathbb{N}} \subseteq \mathbb{K}$ be ensembles of subsets of $\mathbb{F}$ and $\mathbb{K}$, respectively. Let $m = m(k)$ and $\ell = \ell(k)$ be ensembles of integers such that $|H|^m = |G|^\ell = k$.

[22] More generally, for any code C that is locally testable and decodable (such as the LDE code), one can transform an HIP for the language $\mathcal{L}$ into an IPP for the language $C(\mathcal{L}) = \{C(x) : x \in \mathcal{L}\}$. Moreover, if the query location produced by the HIP verifier is uniformly distributed (which is typically the case), then local testability by itself suffices.

Recall that for a given string $x \in \{0,1\}^k$, we define $\mathsf{LDE}_{\mathbb{F},H,m}$ as the unique individual degree $|H|-1$ polynomial $P : \mathbb{F}^m \to \mathbb{F}$ such that $P(z) = x_z$, for every $z \in H^m$ (where we identify the sets H^m and $[k]$ in some, computationally efficient, canonical way). Similarly, we define $\mathsf{LDE}^{\mathbb{K}}_{G,\ell}$ as the unique individual degree $|G|-1$ polynomial $P : \mathbb{K}^\ell \to \mathbb{K}$ such that $P(z) = x_z$, for every $z \in G^\ell$ (where now we identify G^ℓ and $[k]$).

▶ **Lemma 12.** *Let Let Let $\mathbb{F}$ and $\mathbb{K}$ be finite field ensembles as defined above. Let $\mathcal{L} = \cup_{k\in\mathbb{N}}\mathcal{L}_k$ be a language such that $\mathcal{L}_k \subseteq \{0,1\}^k$ for every $k \in \mathbb{N}$. Suppose that $\mathcal{L}$ has a ρ-round* HIP*, with respect to the code $\mathsf{LDE}_{\mathbb{K},G,r}$, with soundness error $\delta = \delta(k) \in [0,1]$ and communication complexity $c = c(k)$. Then, for every parameter $r = r(k) \geq 1$, the language $\mathcal{L}$ also has a $(\rho + r + 1)$-round* HIP*, with respect to the code $\mathsf{LDE}_{\mathbb{F},H,m}$, with soundness error $\left(\delta + O\left(\frac{|H|\cdot m}{|\mathbb{F}|}\right)\right)$ and communication $\left(c + \mathsf{poly}(k^{1/r}, |H|, r, \log|\mathbb{F}|)\right)$.*

Furthermore, the computational overhead for the verifier is $\mathsf{poly}(k^{1/r}, |H|, r, \log|\mathbb{F}|)$ and the computational overhead for the prover is $\mathsf{poly}(k)$.

We remark that for the furthermore part, we make use of the [45] constant-round variant of the GKR protocol.

Proof of Lemma 12. Before presenting the desired HIP, we start with some algebraic notation and basic facts. Throughout this proof we use $\langle\cdot,\cdot\rangle_{\mathbb{K}}$ and $\langle\cdot,\cdot\rangle_{\mathbb{F}}$ to denote inner products over the fields $\mathbb{K}$ and $\mathbb{F}$, respectively.

Recall that elements in $\mathbb{K}$ are represented as vectors in $\mathbb{F}^s$. Let $b_1,\ldots,b_s : \mathbb{K}^* \to \mathbb{F}^*$ be functions defined as follows. For every $\alpha \in \mathbb{K}^*$ it holds that $\alpha = (b_1(\alpha),\ldots,b_s(\alpha))$. That is, the functions $b_1,\ldots,b_s$ decompose a vector $w \in \mathbb{K}^t$ into its s components over $\mathbb{F}^t$.

▶ **Proposition 13.** *For every $w \in \mathbb{K}^k$ and $x \in \{0,1\}^k$ it holds that*

$$\langle w, x\rangle_{\mathbb{K}} = (\langle b_1(w), x\rangle_{\mathbb{F}}, \ldots, \langle b_s(w), x\rangle_{\mathbb{F}}).$$

Proof. We denote by $*$ multiplication in $\mathbb{K}$ and by $\cdot$ multiplication in $\mathbb{F}$. For $k = 1$ the proposition simply states that, for $w \in \mathbb{K}$ and $x \in \{0,1\}$ it holds that $w * x = (b_1(w)\cdot x,\ldots,b_s(w)\cdot x)$. The latter can be easily verified to hold for $x \in \{0,1\}$ by observing that, in both $\mathbb{K}$ and $\mathbb{F}$, multiplication by $x = 0$ always returns 0 and multiplication by $x = 1$ is identity. The proposition follows by induction on k. ◀

We proceed to describe the HIP $(\mathcal{P}',\mathcal{V}')$. Let $(\mathcal{P},\mathcal{V})$ be an HIP for $\mathcal{L}$, with respect to the code $\mathsf{LDE}_{\mathbb{K},G,r}$, with soundness error δ. To prove the lemma, we need to construct an HIP $(\mathcal{P}',\mathcal{V}')$ for $\mathcal{L}$, with respect to the code $\mathsf{LDE}_{\mathbb{F},H,m}$.

First, $\mathcal{P}'$ and $\mathcal{V}'$ emulate the HIP $(\mathcal{P},V)$. If $\mathcal{V}$ rejects, then $\mathcal{V}'$ immediately rejects. Otherwise, $\mathcal{V}$ outputs a pair $(z,\nu) \in \mathbb{K}^\ell \times \mathbb{K}$ with the associated claim that $\left(\mathsf{LDE}^{\mathbb{K}}_{G,\ell}(x)\right)|_z = \nu$. Since $\mathsf{LDE}^{\mathbb{K}}_{G,\ell}$ is a $\mathbb{K}$-linear code, there exists a vector $w \in \mathbb{K}^k$ (that depends only on the code $\mathsf{LDE}^{\mathbb{K}}_{G,\ell}$ and the point z) such that $\left(\mathsf{LDE}^{\mathbb{K}}_{G,\ell}(x)\right)|_z = \langle w, x\rangle$, for every $x \in \{0,1\}^k$. Thus, $\mathcal{V}'$ only needs to verify that $\langle w, x\rangle_{\mathbb{K}} = \nu$.

For every $i \in [s]$, let $w_i \stackrel{\text{def}}{=} b_i(w) \in \mathbb{F}^k$ and let $\nu_i \stackrel{\text{def}}{=} b_i(\nu) \in \mathbb{F}$. By Proposition 13 the $\mathbb{K}$-linear equation $\langle w, x\rangle = \nu$ is equivalent to the following s $\mathbb{F}$-linear equations:

$$\forall i \in [s], \quad \langle w_i, x\rangle_{\mathbb{F}} = \nu_i. \tag{5}$$

The verifier $\mathcal{V}'$ chooses at random an $\mathbb{F}$-linear combination of these s linear equations. Namely, it selects at random $\gamma_1,\ldots,\gamma_s \in \mathbb{F}$ and sends these coefficients to the prover. Let $w' \stackrel{\text{def}}{=} \sum_{i\in[s]} \gamma_i \cdot w_i$ and $\nu' \stackrel{\text{def}}{=} \sum_{i\in[s]} \gamma_i \cdot \nu_i$ (where the summations are over $\mathbb{F}$). Note that if

Equation (5) holds then (with probability 1) $\langle w', x\rangle_{\mathbb{F}} = \nu'$, whereas if Equation (5) does not hold then $\langle w', x\rangle_{\mathbb{F}} \neq \nu'$ with probability $1 - \frac{1}{|\mathbb{F}|}$ over the choice of $\gamma_1, \ldots, \gamma_s \in \mathbb{F}$. We next observe that the latter is an $\mathbb{F}$-linear claim about the input x and such claims can be directly solved using the sumcheck protocol.

Let $\tilde{x} : \mathbb{F}^m \to \mathbb{F}$ (resp., $\tilde{w'}$) be the low degree extension of the input x (resp., the vector $w' \in \mathbb{F}^k$) with respect to the field $\mathbb{F}$, set H and dimension m. That is, $\tilde{x}$ and $\tilde{w'}$ are the unique individual degree $|H| - 1$ polynomial that agree with x and w', respectively, on H^m. Let $P : \mathbb{F}^m \to \mathbb{F}$ be defined as the individual degree $2(|H|-1)$ polynomial $P(z) = \tilde{w'}(z) \cdot \tilde{x}(z)$. Note that $\sum_{z \in H^m} P(z) = \langle w', x\rangle_{\mathbb{F}}$. Thus, checking that $\langle w', x\rangle_{\mathbb{F}} = \nu'$ is equivalent to $\sum_{z \in H^m} P(z) = \nu'$ which we can solve by having the prover and verifier run the sumcheck protocol with respect to the polynomial P.[23]

In case the sumcheck verifier rejects then $\mathcal{V}'$ immediately rejects. Otherwise, the result is a pair $(z'', \nu'') \in \mathbb{F}^m \times \mathbb{F}$. The prover sends to the verifier the value $\mu = \tilde{x}(z'')$. The verifier $\mathcal{V}'$ checks that $\mu \cdot \tilde{w'}(z'') = \nu''$ and if so it outputs (z'', μ), otherwise it rejects. This completes the description of the protocol.

Actually, one point about this protocol remains unclear - how can the verifier efficiently compute $\tilde{w'}(z'')$. If we were to ignore the *computational* resources of the verifier, then we could do this by brute force (e.g., in time roughly $|H|^m$), since $\tilde{w'}$ is independent of the input x. Nevertheless, we do aim for efficient verification and so we need to be able to compute $\tilde{w'}(z'')$ efficiently. We will do so by using additional interaction with the prover, based on the [45] variant of the GKR protocol. We give a sketch in the following paragraph.

Computing $\tilde{w'}(z'')$. We start by taking a closer look at the vector w defined above. By the definition of the low degree extension (see Section 2.3), the vector $w \in \mathbb{K}^{G^\ell}$ is defined as $w_h = \beta(h, z)$, for every $h \in G^\ell$, where β is as defined in Section 2.3. Thus, we have that:

$$\tilde{w'}(z'') = \sum_{i \in [s]} \gamma_i \cdot b_i \left(\sum_{h \in H^m} \beta(h, z'') \cdot \beta(h, z) \right). \tag{6}$$

We observe that Equation (6) can be represented as a (highly uniform) depth $O(\log(s) + m \cdot \log(H) + \log(|G|) + \log(\ell)) = O(\log(k))$ Boolean circuit (on input z, z'') of size $s \cdot H^m \cdot \mathsf{poly}(|G|, \ell, \log(|\mathbb{K}|)) = \mathsf{poly}(k)$. Applying the [45] variant of the GKR protocol, we obtain an r-round interactive proof for verifying Equation (6) in which the verifier runs in time $k^{O(1/r)} \cdot \mathsf{polylog}(|\mathbb{F}|)$ and with similar communication complexity.

Completeness. Fix $x \in \mathcal{L}$. By the completeness of $(\mathcal{P}, \mathcal{V})$, the verifier outputs $(z, \nu) \in \mathbb{K}^\ell \times \mathbb{K}$ such that $\left(\mathsf{LDE}^{\mathbb{K}}_{G,\ell}(x)\right)\big|_z = \nu$, or equivalently, $\langle w, x\rangle_{\mathbb{K}} = \nu$. By Proposition 13 this implies that $\langle w_i, x\rangle_{\mathbb{F}} = \nu_i$, for every $i \in [s]$. Therefore, for every $\gamma_1, \ldots, \gamma_s \in \mathbb{F}$ it holds that:

$$\langle w', x\rangle_{\mathbb{F}} = \sum_{i \in [s]} \gamma_i \cdot \langle w_i, x\rangle_{\mathbb{F}} = \sum_{i \in [s]} \gamma_i \cdot \nu_i = \nu'.$$

By definition of P, this means that $\sum_{z \in H^m} P(z) = \langle w', x\rangle_{\mathbb{F}} - \nu' = 0$ and the completeness of the sumcheck protocol implies that $\nu'' = P(z'') = \tilde{x}(z'') \cdot \tilde{w}(z'')$. Thus the verifier accepts when checking that $\mu \cdot \tilde{w}(z'') = \nu''$.

[23] We remark that while we defined sumcheck as a protocol for the language $\mathcal{L} = \{x \in \mathbb{F}^k : \sum_{i \in [k]} x_i = 0\}$, a trivial, standard modification of the sumcheck protocol yields a protocol for $\mathcal{L}_\nu = \{x \in \mathbb{F}^k : \sum_{i \in [k]} x_i = \nu\}$, for every $\nu \in \mathbb{F}$.

Soundness. Fix $x \notin \mathcal{L}$ and a cheating prover strategy $\mathcal{P}^*$. By the soundness of (P, V), with probability $1 - \varepsilon$, the verifier either rejects (in which case $\mathcal{V}'$ also rejects) or outputs $(z, \nu) \in \mathbb{K}^\ell \times \mathbb{K}$ such that $\langle w, x \rangle_{\mathbb{K}} \neq \nu$. Assuming that the latter holds, by Proposition 13 there exists some $i^* \in [s]$ such that $\langle w_{i^*}, x \rangle_{\mathbb{F}} \neq \nu_{i^*}$. Therefore,

$$\begin{aligned}
\Pr\left[\langle w', x \rangle_{\mathbb{F}} = \nu'\right] &= \Pr\left[\sum_{i \in [s]} \gamma_i \cdot \langle w_i, x \rangle_{\mathbb{F}} = \sum_{i \in [s]} \gamma_i \cdot \nu_i\right] \\
&= \Pr\left[\gamma_{i^*} \cdot (\langle w_{i^*}, x \rangle_{\mathbb{F}} - \nu_{i^*}) = \sum_{i \neq i^*} \gamma_i \cdot (\nu_i - \langle w_i, x \rangle_{\mathbb{F}})\right] \\
&= 1/|\mathbb{F}|.
\end{aligned}$$

Thus, with probability $1 - \frac{1}{|\mathbb{F}|}$ it holds that $\langle w', x \rangle_{\mathbb{F}} \neq \nu'$, and in particular $\sum_{z \in H^m} P(z) \neq 0$ (where P is the polynomial as defined above).

Hence, by the soundness of the sumcheck protocol, with probability $\frac{|H| \cdot m}{|\mathbb{F}|}$ either the sumcheck verifier rejects (in which case we also reject) or it outputs a pair $(z'', \nu'') \in \mathbb{F}^m \times \mathbb{F}$ such that $P(z'') \neq \nu''$, or in other words $\tilde{x}(z'') \cdot \tilde{w}(z'') \neq \nu''$. Now, the prover sends over a value μ. If $\tilde{x}(z'') = \mu$ then, conditioned on the above event, the verifier rejects when checking that $\mu \cdot \tilde{w}(z'') = \nu'$. If $\tilde{x}(z'') \neq \mu$, then the verifier outputs a pair (z'', μ) such that $\left(\mathsf{LDE}_{\mathbb{F},H,m}(x)\right)_{z''} \neq \mu$ as desired. By a union bound, the overall soundness error is $\varepsilon + \frac{1}{|\mathbb{F}|} + \frac{|H| \cdot m}{|\mathbb{F}|}$.

Complexity. On top of the ρ rounds that $(\mathcal{P}, \mathcal{V})$ takes, the verifier also sends the message $(\gamma_1, \ldots, \gamma_s)$, but this message can be appended to the last message from $\mathcal{V}$ to $\mathcal{P}$. In addition, the two parties run an r-round sumcheck protocol and an r round variant of the GKR protocol. There is one additional message from the prover with the value μ, so the overall number of rounds is $\rho + O(r)$.

The communication in the first part of the protocol (i.e., the emulation of (P, V)) is c. In addition, the verifier sends the linear combination $(\gamma_1, \ldots, \gamma_s)$ which takes $s \cdot \log |\mathbb{F}|$ bits. Lastly, both the sumcheck and the GKR protocol add communication $\mathsf{poly}(k^{1/r}, |H|, r, \log |\mathbb{F}|)$ and the additional prover message is just $\log_2 |\mathbb{F}|$ bits.

As for the verifier's complexity, beyond running the original $(\mathcal{P}, \mathcal{V})$ protocol, it runs the sumcheck and GKR protocols which takes time $\mathsf{poly}(k^{1/r}, |H|, r, \log |\mathbb{F}|)$. The prover's additional time in running these two protocols is $\mathsf{poly}(|H|^m) = \mathsf{poly}(k)$. ◀

3.4 Booleanity Testing

In this subsection we show that HIPs can efficiently check that their input is the low-degree extension of a *Boolean* assignment. To do so, we follow an idea of Kalai and Raz [46], which was introduced in the context of constructing interactive PCPs.

We show a simple reduction from checking whether a polynomial $P : \mathbb{F}^m \to \mathbb{F}$ is Boolean valued in a subcube H^m (i.e., $P|_{H^m} H^m \to \{0, 1\}$) to checking whether a related (slightly higher degree) polynomial Q vanishes on H^m. Specifically, consider the polynomial $Q(x) = P(x) \cdot (1 - P(x))$, and observe that P is Boolean-valued in H^m if only if Q is identically zero in H^m. Checking whether a given polynomial is identically 0 (i.e., vanishes) on a subcube of its domain can be solved via a fairly well-known reduction to the sumcheck protocol. We also note that the reduction from P to Q is local (i.e., each query to Q can be computed by a single query to P) and therefore can be used in our setting.

We start by showing an HIP for inputs that vanish on a subcube. We first note that checking whether an individual degree $|H|-1$ polynomial vanishes on the subcube H^m is trivial, since such a polynomial vanishes on H^m if and only if it vanishes on $\mathbb{F}^m$. The actual challenge is checking whether a higher degree polynomial (e.g., with individual degree $|G|-1$ for some G such that $|G|>|H|$) vanishes on H^m.

Formally, for a given field ensemble $\mathbb{F}$, ensembles of subsets $H, G \subseteq \mathbb{F}$ and dimension m, let $\mathsf{Vanishing\text{-}Subcube}_{\mathbb{F},H,m,G}$ be the set of all functions $f : G^m \to \mathbb{F}$ that vanish on H^m (i.e., $f|_{H^m} \equiv 0$).

The following proposition, which gives an HIP for $\mathsf{Vanishing\text{-}Subcube}_{\mathbb{F},H,m,G}$, is implicit in many classical constructions of PCPs (e.g., [5]). We include a proof in Appendix D for completeness.

▶ **Proposition 14.** *Let $\mathbb{F}$ be a constructible field ensemble, let $H \subseteq G \subseteq \mathbb{F}$ be ensembles of subsets, and let $m = m(k)$. For every $r = r(k) \leq \frac{\log(k)}{\log\log(k)}$, there exists an $(r+2)$-round (public-coin) HIP for $\mathsf{Vanishing\text{-}Subcube}_{\mathbb{F},H,m,G}$, with respect to the code $\mathsf{LDE}_{\mathbb{F},G,m}$, with soundness error $O\left(\frac{m\cdot|G|}{|\mathbb{F}|}\right)$ and communication complexity $m\cdot\log(|\mathbb{F}|) + |G|^{\lceil m/r\rceil}\cdot r\cdot\log|\mathbb{F}|$. The verifier runs in time $|G|^{\lceil m/r\rceil}\cdot r\cdot\mathsf{polylog}(|\mathbb{F}|)$ and the prover runs in time $\mathsf{poly}(|\mathbb{F}|^m)$.*

Denote by $\mathsf{Bool}_{\mathbb{F}}$ the set of all Boolean strings, viewed as a subset of $\mathbb{F}^*$. We show an HIP for Bool, which given access to a polynomial $P = \mathsf{LDE}_{\mathbb{F},H,m}(x)$ for some $x \in \mathbb{F}^k$, checks that $x \in \{0,1\}^k$.

▶ **Proposition 15.** *Let $\mathbb{F}$ be a constructible field ensemble, let $H \subseteq \mathbb{F}$, and let $m \in \mathbb{N}$. For every $r \in [m]$, there exists an $(r+2)$-round (public-coin) HIP for $\mathsf{Bool}_{\mathbb{F},H,m}$, with respect to the code $\mathsf{LDE}_{\mathbb{F},H,m}$, with communication complexity $O(r\cdot(2d+|H|-1)^{m/r}\cdot\log|\mathbb{F}| + m\cdot\log(|\mathbb{F}|))$ and soundness error $O\left(\frac{m\cdot|H|}{|\mathbb{F}|}\right)$.*

Proof. Given a degree d polynomial $P : \mathbb{F}^m \to \mathbb{F}$ such that $P = \mathsf{LDE}_{\mathbb{F},H,m}(x)$ for some $x \in \mathbb{F}^{|H|^m}$, define the degree $2d$ polynomial $Q : \mathbb{F}^m \to \mathbb{F}$ as $Q(x) \stackrel{\text{def}}{=} P(x)\cdot(1-P(x))$. Note that we can write $Q = \mathsf{LDE}_{\mathbb{F},G,m}(y)$ for $H \subseteq G \subseteq \mathbb{F}$ and $y \in \mathbb{F}^{|G|^m}$, where $|G| = O(|H|)$.

Observe that P is Boolean-valued in H^m if and only if Q is identically 0 in H^m (this follows from the fact that the univariate polynomial $z\cdot(1-z)$ has exactly two roots: 0 and 1). Thus, to verify that P is Boolean-valued in H^m, we run the HIP for $\mathsf{Vanishing\text{-}Subcube}_{\mathbb{F},H,m,G}$ in Proposition 14, with respect to the polynomial Q. Note that each query $Q(x)$ can be answered by a single query to P (specifically, by returning $P(x)\cdot(1-P(x))$). Correctness follows from the correctness of the HIP for $\mathsf{Vanishing\text{-}Subcube}_{\mathbb{F},H,m,G}$. Communication complexity and soundness error follow from Proposition 14. ◀

4 The Hierarchy Theorem

In this section we prove our main theorem: a round hierarchy for IPPs.

▶ **Theorem 16** (IPP Hierarchy Theorem)**.** *There exists a language $\mathcal{L}$ and a gap function $g(r) = \Theta(r^2)$ such that for every constant $r \geq 1$ it holds that:*

1. **Upper Bound:** *There exists a $g(r)$-round (public-coin) ε-IPP, for $\mathcal{L}$ with communication complexity $n^{O(1/r)}$ and query complexity $\mathsf{poly}(\log n, \varepsilon)$. The verifier runs in time $n^{O(1/r)} + \mathsf{poly}(\log(n), \varepsilon)$ and the prover runs in time $\mathsf{poly}(n)$.*
2. **Lower Bound:** *For every r-round IPP for $\mathcal{L}$, with respect to proximity parameter $\varepsilon = 1/10$, that has query complexity q and communication complexity c, it holds that $\max(c,q) = n^{\Omega(1/r)}$.*

Furthermore, $\mathcal{L}$ also has a $\mathsf{polylog}(n)$*-round (public-coin)* ε-IPP *with communication* $\mathsf{polylog}(n)$ *and query complexity* $\mathsf{poly}(\log n, 1/\varepsilon)$*, and with a* $\mathsf{poly}(\log n, \varepsilon)$*-verifier and* $\mathsf{poly}(n)$*-time prover.*

The O and Ω notation in the theorem statement hide universal constants that do not depend on r. Note that any constant gap between the exponents in the upper and lower bounds can be obtained by increasing g by a suitable constant factor.

The rest of this section is devoted to the proof of Theorem 16. In Section 4.1 we present the language for which we show the IPP round hierarchy, in Section 4.3 we prove the lower bound (see Lemma 23), and in Section 4.2 we prove the upper bound (see Lemma 17). Combining Lemma 23 and Lemma 17 yields Theorem 16.

4.1 The Language: Encoded MOD3

Let $\mathbb{F} = (\mathbb{F}_k)_{k\in\mathbb{N}}$ be a (constructible) field ensemble of characteristic 2 (i.e., each $\mathbb{F}_k$ is an extension field of $\mathsf{GF}(2)$). Let $H = (H_k)_{k\in\mathbb{N}}$ be an ensemble of subsets $H \subseteq \mathbb{F}$ and let $m = m(k)$ be the dimension such that $|H| = \log(k)$, $m = \frac{\log(k)}{\log\log(k)}$ and $|\mathbb{F}| = \Theta(|H|^2 m)$. Denote $n \stackrel{\text{def}}{=} |\mathbb{F}^m|$, and note that $|H|^m = k$ and that $k^2 \leq n \leq k^3$.

We first define an (auxiliary) language $\mathcal{L}_{\mathsf{MOD3}}$, where:

$$\mathcal{L}_{\mathsf{MOD3}} \stackrel{\text{def}}{=} \{x \in \{0,1\}^* \ : \ \mathsf{wt}(x) = 0 \pmod 3\}\,.$$

That is, $\mathcal{L}_{\mathsf{MOD3}}$ simply consists of strings whose Hamming weight is divisible by 3. The actual language for which we prove the IPP lower bound is $\mathsf{Enc\text{-}MOD3} = \mathsf{LDE}_{\mathbb{F},H,m}(\mathcal{L}_{\mathsf{MOD3}})$. That is,

$$\mathsf{Enc\text{-}MOD3} = \left\{\mathsf{LDE}_{\mathbb{F},H,m}(x) \ : \ x \in \mathcal{L}_{\mathsf{MOD3}}\right\}.$$

Or in words, $\mathsf{Enc\text{-}MOD3}$ consists of all m-variate polynomials over $\mathbb{F}$, of individual degree $|H| - 1$, that take Boolean values in H^m such that the integer sum over all elements in H^m is divisible by 3.

4.2 The Upper Bound

In this section, we construct an IPP for $\mathsf{Enc\text{-}MOD3}$. This IPP suffices both for the results in the constant-round regime and poly-logarithmic round regime of Theorem 16.

▶ **Lemma 17.** *For every $r = r(n) \leq \frac{\log(n)}{\log\log(n)}$, there exists an $O(r^2)$-round public-coin ε-IPP for $\mathsf{Enc\text{-}MOD3}$ with perfect completeness and soundness error $1/2$. The communication complexity is $n^{O(1/r)}$ and the query complexity is $\mathsf{poly}(\log(n), 1/\varepsilon)$. Furthermore, the verifier runs in time $\left(n^{O(1/r)} + \mathsf{poly}(\log(n), 1/\varepsilon)\right)$ and the prover runs in time $\mathsf{poly}(n)$.*

The main step in the proof of Lemma 17 is the construction of an HIP for the related language $\mathcal{L}_{\mathsf{MOD3}}$ (defined above), with respect to the LDE code (with the parameters that were specified in Section 4.1). Given this HIP, Lemma 17 follows by using a generic transformation from HIPs (with respect to the LDE encoding) into IPPs, which we establish in Proposition 11.

Before constructing this HIP, as an intermediate goal, we first construct an HIP for $\mathcal{L}_{\mathsf{MOD3}}$, with respect to the low-degree extension with different parameters than those that were set in Section 4.1. Specifically, we shall use a larger field $\mathbb{K}$, whose size is polynomially related to k (rather than poly-logarithmic). In particular, there will be a dependence between the size of $\mathbb{K}$ and the number of rounds in the HIP. Later we will use a generic transformation to

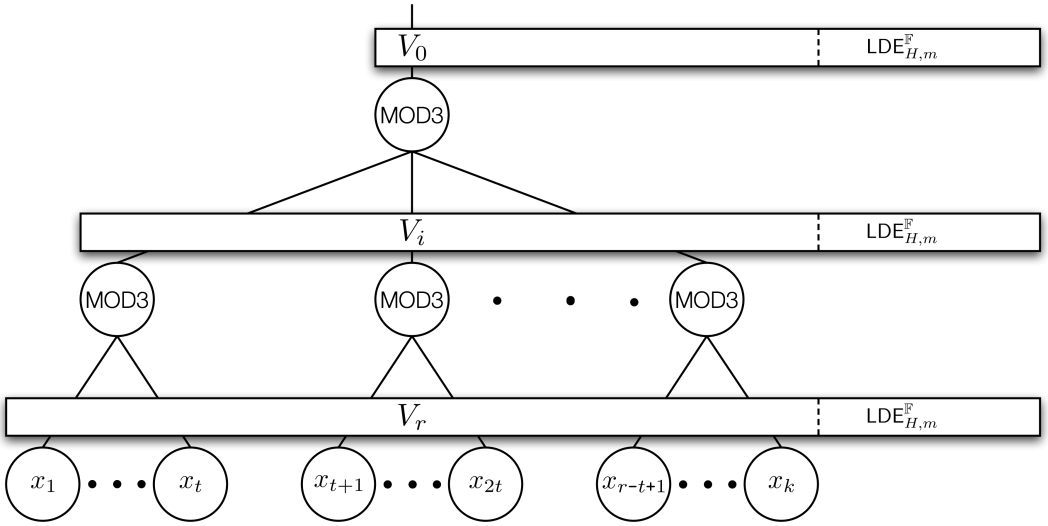


Figure 1 The recursive depth r formula of fan-in $k^{1/r}$ that computes the sum mod 3 of its input $x \in \{0,1\}^k$, and the low-degree extension of each one of the formula's layers when evaluated on x.

convert this HIP into one in which the low degree extension can be over a much smaller field (e.g., of poly-logarithmic size), which in particular does not depend on the number of rounds.

The following lemma, which is the main lemma proved in this section, gives an HIP for $\mathcal{L}_{\mathsf{MOD3}}$ over the relatively large field $\mathbb{K}$.

▶ **Lemma 18.** *Let $r = r(k) \geq 1$, let $\mathbb{K} = (\mathbb{K}_k)_{k\in\mathbb{N}}$ be a constructible field ensemble of size $|\mathbb{K}| = \Omega(r^2 \cdot k^{2/r})$, let $G = (G_k)_{k\in\mathbb{N}} \subseteq \mathbb{K}$ be an ensemble of subsets of $\mathbb{K}$ of size $|G| = k^{1/r}$.*

Then, there exists an r^2-round public-coin HIP *for $\mathcal{L}_{\mathsf{MOD3}}$, with perfect completeness and soundness error $O\left(\frac{r^2 \cdot k^{2/r}}{|\mathbb{K}|}\right)$. The communication complexity is $O(r^2 \cdot k^{2/r} \cdot \log|\mathbb{K}|)$. The verifier runs in time $k^{O(1/r)} \cdot \mathsf{poly}(r, \log(k))$ and the prover runs in time $\mathsf{poly}(|\mathbb{K}|^r)$.*

(See Section 1.2 for a high-level overview of the proof.)

Proof. Let $r = r(k) \geq 1$. Recall that $\mathbb{K} = (\mathbb{K}_k)_{k\in\mathbb{N}}$ is a constructible field ensemble field of size $|\mathbb{K}| = \Omega(r^2 \cdot k^{1/r})$ and that $G = (G_k)_{k\in\mathbb{N}}$ is an ensemble of subsets of size $|G| = k^{1/r}$. Since we only deal with a single input length k (which we think of as varying), in the following we omit the subscripts and use $\mathbb{K}$ (resp., G) when we actually mean $\mathbb{K}_k$ (resp., G_k).

Denote by $t \stackrel{\text{def}}{=} |G| = k^{1/r}$ and fix a canonical ordering $\alpha_1, \ldots, \alpha_t$ of the set of elements in G (i.e., $G = \{\alpha_1, \ldots, \alpha_t\}$). Let $\mathsf{MOD3}_t : \{0,1,2\}^t \to \{0,1,2\}$ be defined as $\mathsf{MOD3}_t(\sigma_1, \ldots, \sigma_t) \stackrel{\text{def}}{=} \sum_{j\in[t]} \sigma_j \pmod 3$.

Fix an input $x \in \{0,1\}^k$. As described in Section 1.2, we define polynomials $V_0, \ldots, V_r$ that contain sums, modulo 3, of certain intervals in x. Taking the [37] view, one can consider a depth r formula, with fan-in $t = k^{1/r}$, composed of $\mathsf{MOD3}_t$ gates, that computes the sum mod 3 of its input (see Figure 1). Viewed this way, each polynomial V_i corresponds to the low degree extension of the i^{th} layer of this formula (counting from output to input).

Since $|G^r| = k$, we can associate elements in G^r with the integers in the set $\{1, \ldots, k\}$ in the natural way. Thus, we can view the input $x \in \{0,1\}^k$ as a function, which we denote by $V_r : G^r \to \{0,1\}$, that is defined as $V_r(p) = x_p$, for every $p \in G^r$. We define functions $V_0, \ldots, V_{r-1}$ via backward recursion as follows. For every $i \in [r]$, let $V_{i-1} : G^{i-1} \to \{0,1,2\}$

be defined as:

$$\forall p \in G^{i-1}, \quad V_{i-1}(p) = \mathsf{MOD3}_t\big(V_i((p,\alpha_1)),\ldots,V_i((p,\alpha_t))\big), \tag{7}$$

where (p,α) denotes the element in G^i which is obtained by concatenating $p \in G^{i-1}$ with $\alpha \in G$. For the case $i = 0$, we define $G^0 = \{\perp\}$, where $\perp$ is defined as the empty string (in particular $(\perp,p) = (p,\perp) = p$), and note that, for $i = 1$, Equation (7) reduces to $V_0(\perp) = \mathsf{MOD3}_t\big(V_1(\alpha_1)\big),\ldots,V_1(\alpha_t)\big)$.

As noted above, intuitively, each V_i should be thought of as specifying a sum of certain intervals in the input, according to a partition (which depends on i). For example, V_r contains the value of each of the individual coordinate of x (i.e., the most fine grained partition) whereas V_0 contains the overall sum (i.e., the coarsest partition). More generally, we have the following immediate fact:

▶ **Fact 19.** *For every $i \in \{0,\ldots,r\}$ and $p \in G^i$ it holds that $V_i(p) = \sum_{q\in G^{r-i}} x_{(p,q)} \pmod 3$ (where (p,q) denotes the concatenation of the two vectors p and q).*

In particular, by setting $i = 0$, we have that $V_0(\perp) = \sum_{q\in G^r} x_q \pmod 3 = \sum_{i\in[k]} x_i \pmod 3$.

For the rest of the proof we use $\tilde{f}$ to denote the low degree extension of a function f (see Section 2.3 for details on the low degree extension encoding) and associate the integers 0, 1 and 2 with three distinct elements in $\mathbb{K}$ in some canonical way (so that we can view $\{0,1,2\} \subseteq \mathbb{K}$). Let $\widetilde{\mathsf{MOD3}}_t : \mathbb{K}^t \to \mathbb{K}$ be the unique individual degree 2 extension of the function $\mathsf{MOD3} : \{0,1,2\}^t \to \{0,1,2\}$ with respect to the field $\mathbb{K}$, the subset $\{0,1,2\} \subseteq \mathbb{K}$, and dimension t. For every $i \in [r]$, let $\tilde{V}_i : \mathbb{K}^i \to \mathbb{K}$ be the unique individual degree $|G| - 1$ extension of V_i with respect to the field $\mathbb{K}$, the set G and dimension i. Let $\tilde{V}_0 \equiv V_0$ (recall that $V_0 : \{\perp\} \to \{0,1,2\}$ is just a singleton value $V_0(\perp) \in \mathbb{K}$). Observe that the polynomial $\tilde{V}_r$ is the low degree extension of the input x with respect to the field $\mathbb{K}$, the set G and dimension r.

A crucial fact that we will use is that, for every $i \in [r]$, each point in $\tilde{V}_{i-1}$ can be expressed as a certain type of composition of the low degree polynomial $\tilde{V}_i$ with the low degree polynomial $\widetilde{\mathsf{MOD3}}$. More specifically, using the properties of the low degree extension (see Section 2.3), it holds that for every $i \in [r]$ and $z \in \mathbb{K}^{i-1}$:

$$\begin{aligned}
\tilde{V}_{i-1}(z) &= \sum_{p\in G^{i-1}} \beta(z,p)\cdot V_{i-1}(p) \\
&= \sum_{p\in G^{i-1}} \beta(z,p)\cdot \mathsf{MOD3}_t\big(V_i((p,\alpha_1)),\ldots,V_i((p,\alpha_t))\big) \\
&= \sum_{p\in G^{i-1}} \beta(z,p)\cdot \widetilde{\mathsf{MOD3}}_t\big(\tilde{V}_i((p,\alpha_1)),\ldots,\tilde{V}_i((p,\alpha_t))\big).
\end{aligned} \tag{8}$$

where the polynomial β is as defined in Section 2.3, and the last equality uses the fact that $\tilde{V}_i|_{G^i} \equiv V_i|_{G^i}$ and $\widetilde{\mathsf{MOD3}}_t|_{\{0,1,2\}^t} \equiv \mathsf{MOD3}_t|_{\{0,1,2\}^t}$.

Using the above definition, we proceed to describe our HIP for $\mathcal{L}_{\mathsf{MOD3}}$. The protocol is performed in r phases, each of which takes at most r rounds of interaction (for a total of at most r^2 rounds). We begin the protocol with a claim about the value of a single point (as a matter of fact, the only point) in $\tilde{V}_0$ (recall that, by Fact 19, the value of $\tilde{V}_0(\perp)$ corresponds to the desired output - the sum modulo 3 of the input bits). In the i^{th} phase, we reduce the task of verifying the value of a single (arbitrary) point in $\tilde{V}_{i-1}$ to verifying the value of a single point in $\tilde{V}_i$. Thus, after r phases, we have reduced the problem of

verifying $\tilde{V}_0(\perp) = \sum_{j\in[k]} x_j \pmod 3$ to verifying a single point in $\tilde{V}_r$, which is the low degree extension of the input x.

Define $z_0 = \perp$ and $\nu_0 = 0$. The original claim is that $\tilde{V}_r(z_0) = \nu_0$. We shall maintain the invariant that for every phase $i \in \{0,\ldots,r\}$, at the end of the i^{th} phase, the prover and verifier both know a vector $z_i \in \mathbb{K}^i$ and a scalar $\nu_i \in \mathbb{K}$ such that the current claim is that $\tilde{V}_i(z_i) = \nu_i$. Thus, the goal of the i^{th} phase is to (interactively) reduce the claim $\tilde{V}_{i-1}(z_{i-1}) = \nu_{i-1}$ to a claim of the form $\tilde{V}_i(z_i) = \nu_i$ (for some z_i and ν_i that are generated during the i^{th} phase):

Phase i.

1. **Reduce to Claim about t Points in $\tilde{V}_i$:** The phase begins with a claim that $\nu_{i-1} = \tilde{V}_{i-1}(z_{i-1})$. By Equation (8) this is equivalent to:

$$\nu_{i-1} = \sum_{p\in G^{i-1}} \beta(z_{i-1},p)\cdot \widetilde{\mathsf{MOD3}}_t\big(\tilde{V}_i((p,\alpha_1)),\ldots,\tilde{V}_i((p,\alpha_t))\big) \tag{9}$$

   We now observe that the right-hand side of Equation (9) corresponds to a sum, over an $(i-1)$-dimensional subcube, of the values of a low degree polynomial. Specifically, denote

$$f_{i-1}(w) = \beta(z_{i-1},w)\cdot \widetilde{\mathsf{MOD3}}_t\big(\tilde{V}_i((w,\alpha_1)),\ldots,\tilde{V}_i((w,\alpha_t))\big),$$

   and observe that f_{i-1} has *total* degree $(t-1)\cdot(i-1)+2t\cdot(t-1)\cdot i \le 3t^2 r$ polynomial. Equation (9) can be rewritten as $\nu_{i-1} = \sum_{p\in G^{i-1}} f_{i-1}(p)$. The prover and verifier run an i-round sumcheck protocol with respect to this equation.

   In case the sumcheck verifier rejects, our verifier immediately rejects. Otherwise, the output of the sumcheck protocol is a (random) point $w_{i-1} \in \mathbb{K}^{i-1}$ and value $\gamma_{i-1} \in \mathbb{K}$ with an associated alleged claim that $\gamma_{i-1} = f_{i-1}(w_{i-1})$.
2. **Query Reduction:** At this point the verifier has a claim regarding the values of t points of $\tilde{V}_i$ (specifically, the claim $\gamma_{i-1} = f_{i-1}(w_{i-1})$ refers to the points $(w_i,\alpha_1),\ldots,(w_i,\alpha_t)$). The goal of this step is to reduce this more elaborate claim to a claim about a *single* point in $\tilde{V}_i$:
   - **a.** The prover sends to the verifier the univariate degree $t-1$ polynomial $P_i : \mathbb{K}\to\mathbb{K}$ defined as $P_i(\eta) = \tilde{V}_i(w_i,\eta)$ (given by its t coefficients).
   - **b.** The verifier receives a degree $t-1$ polynomial Q_i (which is allegedly equal to P_i). The verifier checks that $\gamma_i = \beta(z_{i-1},w_{i-1})\cdot \widetilde{\mathsf{MOD3}}_t\big(Q_i(\alpha_1),\ldots,Q_i(\alpha_t)\big)$. If the check fails then the verifier immediately rejects and halts. Otherwise, the verifier chooses a random field element $\eta_i \in \mathbb{K}$ and sends η_i to the prover.
   - **c.** The claim for the next round is that $\tilde{V}_i(z_i) = \nu_i$, where $\nu_i = P_i(\eta_i)$ and $z_i = (w_i,\eta_i)$.

After all of the r phases are complete, the verifier outputs (z_r,ν_r) and the associated claim is that $\tilde{V}_r(z_r) = \nu_r$. Since $\tilde{V}_r$ is simply the low degree extension of the input x, the latter is a claim about a single point in the low degree extension of the input as required by the definition of an HIP verifier.

Complexity. Since the communication complexity of each sumcheck is $O(r\cdot k^{1/r}\cdot\log|\mathbb{K}|)$, the total communication complexity is $O(r^2\cdot k^{1/r}\cdot\log|\mathbb{K}|)$. As for the round complexity, the i^{th} phase uses a sumcheck of $i \le r$ rounds of interaction. Moreover, each sumcheck concludes with a message from the prover to the verifier so we can "piggyback"and attach the polynomial Q_i to that last message from the prover and send back the value η_i as our response (which is still part of the last round of the sumcheck protocol) so each phase just takes $\le r$ rounds and overall we have $\le r^2$ rounds.

As for computational complexity, in the first step of each phase, the parties invoke a sumcheck protocol in which, by Lemma 10, the verifier runs in time $k^{O(1/r)} \cdot r \cdot \mathsf{polylog}|\mathbb{K}|$, and the prover runs in time $\mathsf{poly}(|\mathbb{K}|^r)$. In the second step of each phase, the prover computes and sends P_i, which clearly can be done in time $\mathsf{poly}(|\mathbb{K}|^r)$, and the verifier computes γ_i, which boils down to evaluating the functions β and $\widetilde{\mathsf{MOD3}}_t$ at a single point, which can be done in time $\mathsf{poly}(t, \log k) = k^{O(1/r)} \cdot \mathsf{poly}(\log |\mathbb{K}|)$ (see Proposition 6 and Appendix E for the time complexity of computing β and $\widetilde{\mathsf{MOD3}}_t$, respectively). The obtain the total running times (for the entire r phases), we multiply the time per phase by r.

Completeness. Perfect completeness follows readily from the construction (and the prefect completeness of the sumcheck protocol).

Soundness. To conclude the proof of Lemma 18 we only need to show that soundness holds. Our analysis follows the soundness analysis in [37, Theorem 3.1].

Fix an input $x \in \{0,1\}^k$ such that $\sum_{i\in[k]} x_i \not\equiv 0 \pmod 3$ (i.e. $x \notin \mathcal{L}_{\mathsf{MOD3}}$) and a cheating strategy $\mathcal{P}^*$. Denote by A the event that the verifier does not reject in the interaction with the prover $\mathcal{P}^*$. For every $i \in \{0, 1, \ldots, r\}$, denote by T_i the event that $\tilde{V}_i(z_i) = \nu_i$. Note that since $\sum_{i\in[k]} x_i \neq 0 \pmod 3$ it holds that the event $\neg T_0$ occurs with probability 1. For every $i \in [r]$, let E_i denote the event that the polynomial Q_i that the prover sent is indeed identical to $P_i(\eta) = \tilde{V}_i(w_i, \eta)$.

Our analysis will be based on the following two claims.

▶ **Claim 20.**

$$\Pr\left[A \wedge E_i \mid \neg T_{i-1}\right] \leq \frac{3t^2 r}{|\mathbb{K}|}.$$

Proof. Assume that the event T_{i-1} occurs. Then, by the soundness of the sumcheck protocol, with probability $\frac{3t^2 r}{|\mathbb{K}|}$ (over the verifier's coins in the sumcheck protocol) it holds that $f_{i-1}(w_{i-1}) \neq \gamma_{i-1}$, or in other words $\beta(z_{i-1}, w_{i-1}) \cdot \widetilde{\mathsf{MOD3}}_t\big(\tilde{V}_i((w_{i-1}, \alpha_1)), \ldots, \tilde{V}_i((w_{i-1}, \alpha_t))\big) \neq \gamma_{i-1}$. If the latter happens and then the prover sends the correct polynomial P_i (i.e., the event E_i occurs) then the verifier immediately rejects in Item 2b. Thus, with probability $1 - \frac{3t^2 r}{|\mathbb{K}}$, either the event $\neg A$ or $\neg E_i$ must occur. ◀

On the other hand:

▶ **Claim 21.**

$$\Pr\left[T_i \mid \neg E_i\right] \leq \frac{t}{|\mathbb{K}|}.$$

Proof. The event $\neg E_i$ implies that the polynomial Q_i sent by the prover differs from the correct polynomial P_i. Since both Q_i and P_i are degree $t-1$ polynomials, they can agree on at most $t-1$ points, and so, with probability $1 - \frac{t-1}{|\mathbb{K}|}$ over the choice of η_i it holds that $\nu_i = Q(\eta_i) \neq P(\eta_i) = \tilde{V}_i((w_i, \eta_i)) = \tilde{V}_i(z_i)$. ◀

Finally, observe that the probability that the verifier errs is simply $\Pr[A \wedge \neg T_r]$, which

we can bound (using Claim 20, Claim 21 and elementary probability theory) as follows:

$$\begin{aligned}
\Pr[A \wedge T_r] &= \Pr[A \wedge \neg T_0 \wedge T_r] \\
&\leq \Pr\left[\exists i \in [r] \text{ such that } A \wedge \neg T_{i-1} \wedge T_i\right] \\
&\leq \sum_{i=1}^{r} \Pr\left[A \wedge \neg T_{i-1} \wedge T_i\right] \\
&= \sum_{i=1}^{r} \left(\Pr[A \wedge \neg T_{i-1} \wedge T_i \wedge E_i] + \Pr[A \wedge \neg T_{i-1} \wedge T_i \wedge \neg E_i]\right) \\
&\leq \sum_{i=1}^{r} \left(\Pr\left[A \wedge E_i \mid \neg T_{i-1}\right] + \Pr\left[T_i \mid \neg E_i\right]\right) \\
&\leq \sum_{i=1}^{r} \left(\frac{3t^2 r}{|\mathbb{K}|} + \frac{t}{|\mathbb{K}|}\right) \\
&\leq \frac{4t^2 r^2}{|\mathbb{K}|}.
\end{aligned}$$

This concludes the proof of Lemma 18. ◀

Lemma 18 provides an r^2-round HIP for $\mathcal{L}_{\mathsf{MOD3}}$, with respect to the code $\mathsf{LDE}_{\mathbb{K},G,r}$, where $\mathbb{K}$ is a field ensemble of size $\Theta\left(r^2 \cdot k^{2/r}\right)$. We now use a general result, which is stated and proved in Section 3, which transforms any such HIP, in which the field $\mathbb{K}$ has small characteristic, into an HIP over the code $\mathsf{LDE}_{\mathbb{F},H,m}$ where the size of the field $\mathbb{F}$ is now only *poly-logarithmic* in k. Specifically, by applying Lemma 12 to the protocol of Lemma 18, and using a field $\mathbb{K}$ which is an extension field of some field $\mathbb{F}$ of size $\mathsf{polylog}(k)$, we obtain the following corollary:

▶ **Corollary 22.** *Let $\mathbb{F} = (\mathbb{F}_k)_{k\in\mathbb{N}}$ be a constructible field ensemble, $H = (H_k)_{k\in\mathbb{N}} \subseteq \mathbb{F}$ be an ensembles of subsets of $\mathbb{F}$ and let $m = m(k)$ be a dimension such that $|H| = \log(k)$, $m = \frac{\log(k)}{\log\log(k)}$ and $|\mathbb{F}| = \Theta(|H| \cdot m)$.*

Then, for every parameter $r = r(k) \leq \frac{\log(k)}{\log\log(k)}$, the language $\mathcal{L}_{\mathsf{MOD3}}$ has an has an $O(r^2)$-round (public-coin) HIP with respect to the code $\mathsf{LDE}_{\mathbb{F},H,m}$ with soundness error $1/2$ and communication complexity $k^{O(1/r)}$. The verifier runs in time $k^{O(1/r)}$ and the prover runs in time $\mathsf{poly}(k)$.

Lemma 17 follows from Corollary 22 by applying Proposition 11, which is a generic transformation from any HIP, over the low degree extension encoding, into an IPP.

4.3 The Lower Bound

▶ **Lemma 23.** *Let $r = r(k) \geq 1$ be a constant. For every r-round IPP for Enc-MOD3, with respect to proximity parameter $\varepsilon = 1/10$, with query complexity q and communication complexity $c \geq \Omega(\log n)$, it holds that $\max(c, q) = n^{\Omega(1/r)}$.*

We remark that our proof of Lemma 23 gives a similar result even for super constant values of r (as long as $r = O\left(\sqrt{\frac{\log(n)}{\log\log(n)}}\right)$) but for simplicity we restrict ourselves to constant r. We also remark that the constants in the lemma's statement can be improved but we avoid optimizing them for sake of readability.

Proof. Throughout the proof of Lemma 23 all proofs of proximity refer to proximity parameter $\varepsilon = 1/10$.

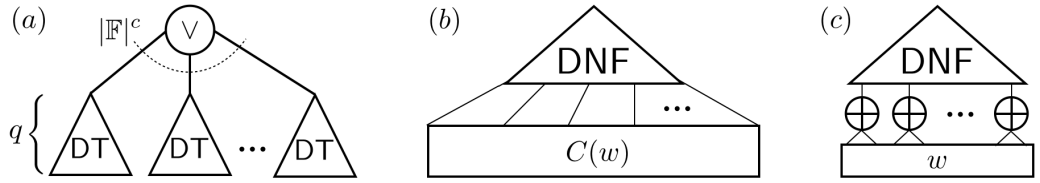


Figure 2 After fixing the randomness, an AMP for Π can be expressed as follows: (a) a disjunction over $O(2^c)$ decision trees of depth $q \cdot \log(|\mathbb{F}|)$, (b) a DNF formula with $O(2^{c+q\cdot\log(|\mathbb{F}|)})$ clauses of width $q \cdot \log|\mathbb{F}|$ over the linear code $C(w)$, and (c) a $\mathsf{DNF}_\oplus$ circuit of size $\tilde{O}(2^{c+q\cdot\log(|\mathbb{F}|)})$ over x.

The following proposition, due to [58] (building on [8, 39, 36]), shows that to prove Lemma 23, it suffices to prove a lower bound for AMPs, which are *public-coin* IPPs with only a *single* round of interaction between the verifier and prover. More precisely, in an AMP for a language $\mathcal{L}$, the verifier first sends a random string r to the prover, who responds with a proof π, which can depend on both the input x and the verifier's message r. Then, given π (and based on its original random coins r), the verifier needs to decide whether to accept or reject. (Note that the verifier is not allowed to toss additional coins after receiving the message from the prover.)

▶ **Proposition 24** (IPP to AMP). *If there exists an r-round (public or private coin)* IPP *for a language $\mathcal{L}$, with communication complexity $c \geq \log(n)$ and query complexity q, then there exists an* AMP *for $\mathcal{L}$ with communication complexity $c^{r+2} \cdot (\log(c) \cdot r)^{O(r)}$ and query complexity $c^{r+1} \cdot q \cdot (\log(c) \cdot r)^{O(r)}$.*

The proof of Proposition 24, which appears in [58, Section 4], proceeds by observing that the private-coin to public-coin transformation of [39] as well as the round reduction transformation of [8, 36], which are transformations on standard interactive proofs, can be applied to IPPs as well.

Thus, given Proposition 24, and using the fact that r is constant, to prove Lemma 23 it suffices to show that every AMP for Enc-MOD3 with query complexity q and communication complexity c satisfies $\max(c, q) = n^{\Omega(1)}$, or equivalently, since $n = O(k^3)$, that $\max(c, q) = k^{\Omega(1)}$. The following proposition, which is inspired by the [58] lower bound, shows that AMPs for properties of *linear codes* can be viewed as distributions over (relatively) small DNF*s of parities*. By DNF of parities, we refer to depth 3 circuits whose bottom layer consists of parity gates, middle layer consists of AND gates and top layer is a single OR gate. In the following we denote such circuits by $\mathsf{DNF}_\oplus$.

▶ **Proposition 25.** *Let $\mathbb{F}$ be an extension field of* GF(2)*, let $C : \mathbb{F}^k \to \mathbb{F}^n$ be an $\mathbb{F}$-linear code, and let $f : \{0,1\}^k \to \{0,1\}$. If there exists an* AMP *for $\Pi_f \stackrel{\text{def}}{=} \{C(x) : x \in \{0,1\}^k \wedge f(x) = 1\}$ with communication complexity $c \geq \log(n)$ and query complexity q, then there exists a distribution $\mathcal{D}$ over $\mathsf{DNF}_\oplus$ circuits of size $2^{O(c+q\cdot\log_2(|\mathbb{F}|))}$ such that $\Pr_{\varphi\sim\mathcal{D}}[\varphi(x) = f(x)] \geq 0.9$, for all $x \in \{0,1\}^k$.*

Proof. Let $\mathcal{V}$ be an AMP verifier for Π. We assume without loss of generality that $\mathcal{V}$ has soundness error at most 0.1 (e.g., by repeating the protocol in parallel $O(1)$ times). Recall that in an AMP protocol, for a given input $y \in \mathbb{F}^n$, the verifier first sends a random string r, then the prover replies with an alleged proof $\pi = \pi(r, y)$, and finally the verifier makes queries to y and decides whether to accept or reject. Denote by $\mathcal{V}^y_{r,\pi}$ the output of the verifier for a fixed string r, given oracle access to y and direct access to π.

For a fixed string r and alleged proof π, the verifier $\mathcal{V}^y_{r,\pi}$ can be represented as an $|\mathbb{F}|$-ary decision tree of depth q (on input y), which we denote by $D_{r,\pi} : \{0,1\}^k \to \{0,1\}$. The completeness and soundness requirements of an AMP guarantee that for a fixed string r, the verifier accepts an input $C(x)$ if and only if there exists a string π such that $\mathcal{V}^{C(x)}_{r,\pi} = 1$. Thus, $\mathcal{V}^{C(x)}_{r,\pi} = \bigvee_{\pi\in\{0,1\}^{O(c)}} D_{r,\pi}\big(C(x)\big)$ (see Figure 2(a)). Observe that by viewing elements of $\mathbb{F}$ in their bit-representation and assigning a clause for each accepting leaf in the decision tree, each $D_{r,\pi}$ can be represented as a *binary* DNF formula with $|\mathbb{F}|^{O(q)}$ clauses of width $O(q \cdot \log|\mathbb{F}|)$. Merging the two consecutive layers of disjunctions, we obtain a binary DNF formula that on input $y \in \mathbb{F}^n$ computes $\mathcal{V}^y_{r,\pi}$ with $2^{O(c+q\cdot\log_2(|\mathbb{F}|))}$ clauses of width $O(q \cdot \log_2(|\mathbb{F}|))$ each (see Figure 2(b) for an illustration).

We next observe that every linear combination over the field $\mathbb{F}$, which is an extension field of $\mathsf{GF}(2)$, can be represented by $\log_2(|\mathbb{F}|)$ linear combinations over $\mathsf{GF}(2)$.[24] Thus, we can view the function $C : \mathbb{F}^k \to \mathbb{F}^n$, which is an $\mathbb{F}$-linear function, as a $\mathsf{GF}(2)$-linear function $C : \mathsf{GF}(2)^{k\cdot\log_2(|\mathbb{F}|)} \to \mathsf{GF}(2)^{n\cdot\log_2(|\mathbb{F}|)}$. Hence, for every random string r, there exists a $\mathsf{DNF}_\oplus$ circuit of size:

$$2^{O(c+q\cdot\log_2(|\mathbb{F}|))} \cdot q \cdot \log_2(|\mathbb{F}|) + n \cdot \log_2(|\mathbb{F}|) = 2^{O(c+q\cdot\log(|\mathbb{F}|))}$$

(which is constructed by composing the code C with the DNF $\bigvee_{\pi\in\{0,1\}^{O(c)}} D_{r,\pi}$) that on input $x \in \{0,1\}^k$ outputs 1 if and only if there exists a proof π that $\mathcal{V}$ would accept, given input $C(x)$.

Therefore, there exists a distribution $\mathcal{D}$ over $\mathsf{DNF}_\oplus$s of size $2^{O(c+q\cdot\log(|\mathbb{F}|))}$ such that for every $x \in \{0,1\}^k$, it holds that $\Pr_{\varphi\in\mathcal{D}}[\varphi(x) = f(x)] \geq 0.9$. This concludes the proof of Proposition 25. ◀

Let $f_{\mathsf{MOD3}} : \{0,1\}^k \to \{0,1\}$ such that $f_{\mathsf{MOD3}} = 1$ if and only if $\sum_{i\in[k]} x_i \equiv 0 \pmod 3$. By Proposition 25, choosing $\Pi = \mathsf{Enc\text{-}MOD3}$, $C = \mathsf{LDE}^{\mathbb{F}}_{H,m}$, $f = f_{\mathsf{MOD3}}$, and using the (easy direction of) Yao's minimax principle, it suffices to show that there exists a distribution $\mathcal{X}$ over inputs in $\{0,1\}^k$ such that for every $\mathsf{DNF}_\oplus$ φ of size $\left(2^{O(c+q\cdot\log_2(|\mathbb{F}|))}\right)$ it holds that $\Pr_{x\in\mathcal{X}}[\varphi(x) = f_{\mathsf{MOD3}}(x)] < 0.9$ (where recall that $|\mathbb{F}| = \mathsf{polylog}(k)$). To that end, we shall use the celebrated result of Razborov [55] and Smolensky [62].

▶ **Theorem 26** (Razborov-Smolensky (see also [68, Theorem 2])). *Every* $\mathsf{AC}^0(\oplus)$ *circuit* φ *of size* s *and depth* d *satisfies*

$$\Pr_{x\in\{0,1\}^k}[\varphi(x) = f_{\mathsf{MOD3}}(x)] < \frac{2}{3} + O\left(\frac{\log(s)^d}{\sqrt{k}}\right).$$

This concludes the proof of Lemma 23. ◀

5 Implications for Classical Interactive Proofs

In this section, we derive from our hierarchy theorem implications to standard interactive proofs (in which the verifier can run in polynomial time). Loosely speaking, in Section 5.1 we show that the round reduction of public-coin interactive proofs, due to Babai and Moran [8], is (almost) optimal among all blackbox transformations, and in Section 5.2 we show that any proof that $\#\mathcal{P} \subseteq \mathcal{AM}$ will require using non-algebrizing techniques.

[24] Fix a linear combination $\alpha \in \mathbb{F}^t$ over $\mathbb{F}$ (the extension field). For every $i \in [\log_2(|\mathbb{F}|)]$, the function $\ell_{\alpha,i}(x) = \mathsf{bit}_i(\langle\alpha, x\rangle)$ that outputs the t^{th} bit of $\langle\alpha, x\rangle$ is a linear function over $\mathsf{GF}(2)$.

5.1 Blackbox Round Reduction Transformations

Babai and Moran [8] proved a "speedup" theorem, which loosely speaking, shows that very r-round public-coin interactive proof protocol can be transformed into an $(r-1)$-round protocol at the cost of increasing the communication complexity quadratically (some quantitative improvements were later obtained by Goldreich, Vadhan and Wigderson [36]). Combined with the private-coin to public-coin transformation of Goldwasser and Sipser [39], one can obtain a similar "speedup" theorem for private-coin interactive proofs.

Vadhan [66] considered the affect of certain transformations on interactive proofs. He introduced the notion of a "blackbox transformation" (defined below) and showed that the aforementioned private-coin to public-coin transformation, and a transformation from 2-sided error to 1-sided error of Goldreich, Mansour and Sipser [32], are (in a certain sense) optimal amongst all *black-box* transformation.

In this section, we use our hierarchy theorem to derive a similar result for the round reduction theorem of Babai and Moran. Following [66], we define a black-box transformation on interactive proofs as a procedure that takes as input an interactive proof $(\mathcal{P},\mathcal{V})$ for some language $\mathcal{L}$ and outputs a new interactive proof $(\mathcal{P}',\mathcal{V}')$, for the same language $\mathcal{L}$, such that:

- The strategy of the verifier $\mathcal{V}'$ can be implemented by an algorithm given oracle access to the strategy of $\mathcal{V}$.
- The strategy of the prover $\mathcal{P}'$ can be implemented by a algorithm given oracle access to the strategy of both $\mathcal{P}$ and $\mathcal{V}$.

Here, the strategy of a party (i.e., prover or verifier) is the function that takes the party's random coins and the history of messages exchanged and outputs its next message. We stress that the new strategies $(\mathcal{P}',\mathcal{V}')$ cannot even explicitly look at the input x; their only access to the input x is given by queries to the strategies $(\mathcal{P},\mathcal{V})$.

An r-to-r' blackbox round reduction transformation, for $r'<r$, is a black-box transformation that, given as input an r-round interactive proof, produces an r'-interactive proof (for the same language). We remark that the [8] round-reduction is a blackbox round reduction transformation, and we show that it is nearly optimal, out of all blackbox reductions.

▶ **Theorem 27.** *There exists a language $\mathcal{L}$ such that for every constant $r\geq 1$, there exists an r-round (public-coin) interactive proof $(\mathcal{P},V)$ for $\mathcal{L}$, with communication complexity $c=c(n)$, such that for every r-to-r' blackbox round reduction transformation T, in the resulting interactive proof $(\mathcal{P}',\mathcal{V}')=T(\mathcal{P},\mathcal{V})$ it holds that either the communication is at least $c^{\Omega(\sqrt{r}/r')}$ or $\mathcal{V}'$ invokes V at least $c^{\Omega(\sqrt{r}/r')}$ times.*

Proof. Let $r\in\mathbb{N}$ be a constant, and consider the language

$$\mathcal{L}_{\mathsf{MOD3}}=\left\{x\in\{0,1\}^k \;:\; \mathsf{wt}(x)=0 \pmod 3\right\}_{k\in\mathbb{N}}.$$

Fix input length $k\in\mathbb{N}$, field $\mathbb{F}$, subset $H\subset\mathbb{F}$, and dimension $m=\frac{\log(k)}{\log\log(k)}$ such that $|H|=\log(k)$ and $|\mathbb{F}|=\Theta(|H|^2 m)$.

By Corollary 22, there exists an r-round HIP for $\mathcal{L}_{\mathsf{MOD3}}$, with respect to the code $\mathsf{LDE}_{\mathbb{F},H,m}$, with communication complexity $c\stackrel{\text{def}}{=}k^{O(1/\sqrt{r})}$. As noted in Proposition 9, this HIP implies an interactive proof $(\mathcal{P},\mathcal{V})$ for $\mathcal{L}_{\mathsf{MOD3}}$, with communication complexity c. Recall that on input $x\in\{0,1\}^k$, the parties $(\mathcal{P},\mathcal{V})$ invoke the HIP for $\mathcal{L}_{\mathsf{MOD3}}$, and the verifier checks the HIP's output claim by computing a single point of $\mathsf{LDE}_{\mathbb{F},H,m}(x)$.

Let T be an r-to-r' blackbox round reduction transformation on interactive proofs, and let $(\mathcal{P}',\mathcal{V}')=T(\mathcal{P},\mathcal{V})$ be the resulting r'-round interactive proof for $\mathcal{L}_{\mathsf{MOD3}}$. Using $(\mathcal{P}',\mathcal{V}')$,

we construct an r'-round ε-IPP for the language

$$\mathsf{Enc\text{-}MOD3} = \{\mathsf{LDE}_{\mathbb{F},H,m}(x) \ : \ x \in \mathcal{L}_{\mathsf{MOD3}}\}\,.$$

Recall that $\mathcal{V}$ only computes $\mathsf{LDE}_{\mathbb{F},H,m}(x)$ and queries it at a single point, and so each oracle call to $\mathcal{V}$ that $\mathcal{V}'$ makes can be emulated by making a single query to $\mathsf{LDE}_{\mathbb{F},H,m}(x)$. Therefore, we can view $(\mathcal{P}', \mathcal{V}')$ as an HIP, with respect to $\mathsf{LDE}_{\mathbb{F},H,m}$, for $\mathcal{L}_{\mathsf{MOD3}}$, with communication complexity c.

By applying Proposition 11 on $(\mathcal{P}', \mathcal{V}')$, we obtain an r'-round IPP for Enc-MOD3; denote its communication complexity by C and query complexity by Q. Finally, by Lemma 23 we have that:

$$\max(C, Q) = k^{\Omega(1/r')} = c^{\Omega(\sqrt{r}/r')}. \qquad \blacktriangleleft$$

5.2 The Algebrization Barrier

The *relatization* framework, introduced by Baker, Gill, and Solovay [9], tried to capture the intuition that we not understand how circuits operate and therefore we may as well treat them as black-boxes. Later on, the seminal result of [52, 60] showed that even without understanding how circuits operate, we can still do more than just evaluate them (i.e., treat them as oracles). Specifically, arithmetizing the circuit, allows us to evaluate points in a *low degree extension* of the function computed by the circuit. The latter cannot be done only via oracle access and has turned out to be incredibly useful.

The *algebrization* framework, introduced by Aaronson and Wigderson [1], tries to capture this additional power. Specifically, in this framework, rather than just giving oracle access to the given function, we give oracle access also to a low degree extension of the function. Results such as $\mathsf{IP} = \mathsf{PSPACE}$ can be showed to have "algebrizing" proofs. Despite the power that we obtain by having access to the low degree extension of the function, [1] also showed that some central questions in complexity theory cannot be proved within this framework (i.e., by "algebrizing") techniques.

Loosely speaking, for two complexity classes $\mathcal{C}_1$ and $\mathcal{C}_2$, the inclusion $\mathcal{C}_1 \subseteq \mathcal{C}_2$ is said to algebrize if $\mathcal{C}_1^A \subseteq \mathcal{C}_2^{\tilde{A}}$ for every oracle A and every low-degree extension $\tilde{A}$ of A. (See [1] for the precise definition, discussions and many more details.) We say that proving the inclusion $\mathcal{C}_1 \subseteq \mathcal{C}_2$ requires non-algebrizing techniques, or cannot be proved via algebrizing techniques, if the inclusion does *not* algebrize.

Before stating our results, we point out that there is an intimate connection (or a high level equivalence) between the class the algebrized class $\mathsf{IP}^{\tilde{A}}$ (where $\tilde{A}$ is the low degree extension of some oracle A) and the notion of HIPs (with respect to the low degree extension encoding). Indeed, in both cases the verifier needs to verify a property of some string, given oracle access to its low degree extension and interaction with the prover. For an $\mathsf{IP}^{\tilde{A}}$ the string is the truth table of A and for HIPs the string is simply the input.

Using this relation, we use our hierarchy theorem to show that the inclusion $\#\mathcal{P} \subseteq \mathcal{AM}$, which is widely believed *not* to hold[25], cannot be proved via algebrizing techniques. As a matter of fact, the proof of Theorem 28 can be easily extended to show that even the containment of $\#\mathcal{P}$ in a powerful variant of $\mathcal{AM}$ in which, for inputs of length N, the verifier is allowed to run in time $2^{o(N)}$ and with $2^{o(N)}$ communication, cannot be proved via algebrizing techniques.

[25] In particular it implies the collapse of the polynomial hierarchy to its second level.

▶ **Theorem 28.** *There exists an oracle A and a low-degree extension $\tilde{A}$ of A such that $\#\mathcal{P}^A \not\subseteq \mathcal{AM}^{\tilde{A}}$.*

Proof Sketch. Consider the problem #CSAT, which is the problem of counting the number of satisfying assignments of a given (Boolean) circuit C, and recall that #CSAT is $\#\mathcal{P}$-complete. Let $A : \{0,1\}^N \to \{0,1\}$ be an oracle and consider an input circuit C that, given as input $x \in \{0,1\}^N$, just outputs $A(x)$. We associate A with its truth table, which is a string of length 2^N. Let $\tilde{A} = \mathsf{LDE}_{\mathbb{F},H,m}(A)$, where $\mathbb{F}, H, m$ are defined as in Section 4.1, with respect to the parameter $k = 2^N$.

Observe that if $\#\mathcal{P}^A \subset \mathcal{AM}^{\tilde{A}}$, then there exists an $\mathcal{AM}$ proof system for computing the number of satisfying assignments of the circuit C, which is exactly the Hamming weight of A (viewed as an k-bit string), in which the communication complexity is $\mathsf{poly}(N)$ and in which the verifier only makes $\mathsf{poly}(N)$ oracle queries to $\tilde{A}$. Thus, following Proposition 11, we can obtain from this $\mathcal{AM}$ proof system a 1-round IPP for Enc-MOD3 with communication and query complexities $\mathsf{poly}(N) = \mathsf{polylog}(k)$, which violates the lower bound in Lemma 23. ◀

▶ Remark (Using Prime Order Fields). We remark that the proof of Theorem 28 is strongly based on the fact that we take a low degree extension over a field that has characteristic 2. Our result can extend to other constant size characteristics but we do not know how to extend it to arbitrary fields. In fact, it is consistent with our result (however unlikely) that there is a proof that $\#\mathcal{P} \subseteq \mathcal{AM}$ based (in a crucial way) on taking the low degree extension of the circuit with respect to a large prime order field.

We remark that we are unaware of any complexity class containments in the literature that are only known based on algebrization using *prime* order fields.[26]

Acknowledgements. We are grateful to Guy Rothblum for sharing [45] with us. We thank Oded Goldreich for his encouragement and support, for many technical and conceptual discussions on the contents of this work, and for valuable comments on its presentation. We thank Justin Holmgren for the proof of Proposition 33. Finally, we wish to thank Shafi Goldwasser, Guy Rothblum, and Rocco Servedio for insightful conversations.

References

1 Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *ACM Trans. Comput. Theory*, 1:2:1–2:54, February 2009. doi:10.1145/1490270.1490272.

2 William Aiello, Shafi Goldwasser, and Johan Håstad. On the power of interaction. *Combinatorica*, 10(1):3–25, 1990. doi:10.1007/BF02122692.

3 Miklós Ajtai, János Komlós, and Endre Szemerédi. An O(n log n) sorting network. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 1–9, 1983. doi:10.1145/800061.808726.

4 Sanjeev Arora and Madhu Sudan. Improved low-degree testing and its applications. *Combinatorica*, 23(3):365–426, 2003. doi:10.1007/s00493-003-0025-0.

5 László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *STOC*, pages 21–31, 1991. doi:10.1145/103418.103428.

[26] The original proof of the IP = PSPACE theorem by Shamir [60] does use prime order fields in an important way, however, the more recent proof of the same result by Goldwasser *et al.* [37] can be based on fields of arbitrary characteristic (see also [53] that gives a proof based on general tensor codes).

6 László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991. doi:10.1007/BF01200056.

7 Laszlo Babai, Peter Frankl, and Janos Simon. Complexity classes in communication complexity theory. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, pages 337–347, Washington, DC, USA, 1986. IEEE Computer Society. doi:10.1109/SFCS.1986.15.

8 László Babai and Shlomo Moran. Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36(2):254–276, 1988.

9 Theodore Baker, John Gill, and Robert Solovay. Relativizations of the P=?NP question. *SIAM Journal on computing*, 4(4):431–442, 1975.

10 Marco L. Carmosino, Jiawei Gao, Russell Impagliazzo, Ivan Mihajlin, Ramamohan Paturi, and Stefan Schneider. Nondeterministic extensions of the strong exponential time hypothesis and consequences for non-reducibility. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 261–270, 2016. doi:10.1145/2840728.2840746.

11 Amit Chakrabarti, Graham Cormode, Navin Goyal, and Justin Thaler. Annotations for sparse data streams. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 687–706. SIAM, 2014.

12 Amit Chakrabarti, Graham Cormode, and Andrew Mcgregor. Annotations in data streams. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming: Part I*, ICALP '09, pages 222–234, Berlin, Heidelberg, 2009. Springer-Verlag. doi:10.1007/978-3-642-02927-1_20.

13 Amit Chakrabarti, Graham Cormode, Andrew McGregor, Justin Thaler, and Suresh Venkatasubramanian. Verifiable stream computation and Arthur–Merlin communication. In *30th Conference on Computational Complexity (CCC 2015)*, volume 33, pages 217–243. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015.

14 Richard Chang, Benny Chor, Oded Goldreich, Juris Hartmanis, Johan Håstad, Desh Ranjan, and Pankaj Rohatgi. The random oracle hypothesis is false. *Journal of Computer and System Sciences*, 49(1):24–39, 1994.

15 Gil Cohen, Ivan Bjerre Damgård, Yuval Ishai, Jonas Kölker, Peter Bro Miltersen, Ran Raz, and Ron D. Rothblum. Efficient multiparty protocols via log-depth threshold formulae - (extended abstract). In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*, pages 185–202, 2013. doi:10.1007/978-3-642-40084-1_11.

16 Graham Cormode, Michael Mitzenmacher, and Justin Thaler. Practical verified computation with streaming interactive proofs. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 90–112. ACM, 2012.

17 Graham Cormode, Michael Mitzenmacher, and Justin Thaler. Streaming graph computations with a helpful advisor. *Algorithmica*, 65(2):409–442, 2013.

18 Samira Daruki, Justin Thaler, and Suresh Venkatasubramanian. Streaming verification in data analysis. *arXiv preprint arXiv:1509.05514*, 2015.

19 Funda Ergün, Ravi Kumar, and Ronitt Rubinfeld. Fast approximate probabilistically checkable proofs. *Inf. Comput.*, 189(2):135–159, 2004. doi:10.1016/j.ic.2003.09.005.

20 Eldar Fischer, Yonatan Goldhirsh, and Oded Lachish. Partial tests, universal tests and decomposability. In *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 483–500, 2014. doi:10.1145/2554797.2554841.

21 Eldar Fischer, Oded Lachish, and Yadu Vasudev. Trading query complexity for sample-based testing and multi-testing scalability. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 1163–1182. IEEE, 2015.

22 Lance Fortnow and Michael Sipser. Are there interactive protocols for CO-NP languages? *Inf. Process. Lett.*, 28(5):249–251, 1988. doi:10.1016/0020-0190(88)90199-8.

23 Peter Gemmell and Madhu Sudan. Highly resilient correctors for polynomials. *Inf. Process. Lett.*, 43(4):169–174, 1992. doi:10.1016/0020-0190(92)90195-2.

24 Oded Goldreich. *Computational complexity - a conceptual perspective*. Cambridge University Press, 2008.

25 Oded Goldreich. Valiant's polynomial-size monotone formula for majority. Unpublished, 2011. URL: http://www.wisdom.weizmann.ac.il/~oded/PDF/mono-maj.pdf.

26 Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017. URL: http://www.wisdom.weizmann.ac.il/~oded/pt-intro.html.

27 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998.

28 Oded Goldreich and Tom Gur. Universal locally testable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:42, 2016. URL: http://eccc.hpi-web.de/report/2016/042.

29 Oded Goldreich and Tom Gur. Universal locally verifiable codes and 3-round interactive proofs of proximity for CSP. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:192, 2016. URL: http://eccc.hpi-web.de/report/2016/192.

30 Oded Goldreich, Tom Gur, and Ilan Komargodski. Strong locally testable codes with relaxed local decoders. In *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, pages 1–41, 2015. doi:10.4230/LIPIcs.CCC.2015.1.

31 Oded Goldreich, Tom Gur, and Ron D. Rothblum. Proofs of proximity for context-free languages and read-once branching programs - (extended abstract). In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 666–677, 2015. doi:10.1007/978-3-662-47672-7_54.

32 Oded Goldreich, Yishay Mansour, and Michael Sipser. Interactive proof systems: Provers that never fail and random selection (extended abstract). In *28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987*, pages 449–461, 1987. doi:10.1109/SFCS.1987.35.

33 Oded Goldreich and Dana Ron. On sample-based testers. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:109, 2013. URL: http://eccc.hpi-web.de/report/2013/109.

34 Oded Goldreich and Dana Ron. On learning and testing dynamic environments. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:29, 2014. URL: http://eccc.hpi-web.de/report/2014/029/.

35 Oded Goldreich and Madhu Sudan. Locally testable codes and PCPs of almost-linear length. *J. ACM*, 53(4):558–655, 2006. doi:10.1145/1162349.1162351.

36 Oded Goldreich, Salil P. Vadhan, and Avi Wigderson. On interactive proofs with a laconic prover. *Computational Complexity*, 11(1-2):1–53, 2002. doi:10.1007/s00037-002-0169-0.

37 Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: interactive proofs for muggles. In *STOC*, pages 113–122, 2008.

38 Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989. doi:10.1137/0218012.

39 Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 59–68. ACM, 1986.

40 Mika Göös, Toniann Pitassi, and Thomas Watson. The landscape of communication complexity classes. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:49, 2015. URL: http://eccc.hpi-web.de/report/2015/049.

41 Mika Göös, Toniann Pitassi, and Thomas Watson. Zero-information protocols and unambiguity in Arthur-Merlin communication. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 113–122, 2015. doi:10.1145/2688073.2688074.

42 Tom Gur and Ran Raz. Arthur-Merlin streaming complexity. *Information and Computation*, 243:145–165, 2015. doi:10.1016/j.ic.2014.12.011.

43 Tom Gur and Ron D. Rothblum. Non-interactive proofs of proximity. *Computational Complexity*, pages 1–109, 2016. doi:10.1007/s00037-016-0136-9.

44 Shlomo Hoory, Avner Magen, and Toniann Pitassi. Monotone circuits for the majority function. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2006 and 10th International Workshop on Randomization and Computation, RANDOM 2006, Barcelona, Spain, August 28-30 2006, Proceedings*, pages 410–425, 2006. doi:10.1007/11830924_38.

45 Yael Kalai and Guy N. Rothblum. Constant-round interactive proofs for NC^1. Unpublished observation, 2009.

46 Yael Tauman Kalai and Ran Raz. Interactive PCP. In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, pages 536–547, 2008. doi:10.1007/978-3-540-70583-3_44.

47 Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. Delegation for bounded space. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 565–574, 2013. doi:10.1145/2488608.2488679.

48 Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. How to delegate computations: the power of no-signaling proofs. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 485–494, 2014. doi:10.1145/2591796.2591809.

49 Yael Tauman Kalai and Ron D. Rothblum. Arguments of proximity - [extended abstract]. In *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 422–442, 2015. doi:10.1007/978-3-662-48000-7_21.

50 Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *STOC*, pages 723–732, 1992.

51 Hartmut Klauck. On Arthur Merlin games in communication complexity. In *Computational Complexity (CCC), 2011 IEEE 26th Annual Conference on*, pages 189–199. IEEE, 2011.

52 Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992. doi:10.1145/146585.146605.

53 Or Meir. IP = PSPACE using error-correcting codes. *SIAM J. Comput.*, 42(1):380–403, 2013. doi:10.1137/110829660.

54 Ran Raz, Gábor Tardos, Oleg Verbitsky, and Nikolai Vereshagin. Arthur-Merlin games in boolean decision trees. In *Computational Complexity, 1998. Proceedings. Thirteenth Annual IEEE Conference on*, pages 58–67. IEEE, 1998.

55 A. Razborov. Lower bounds for the size of circuits of bounded depth with basis $\{\wedge, \oplus\}$. Notes of the Academy of Science of the USSR: 41(4) : 333-338, 1987.

56 Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 49–62, 2016. doi:10.1145/2897518.2897652.

57 Guy N. Rothblum. *Delegating computation reliably: paradigms and constructions*. PhD thesis, Massachusetts Institute of Technology, 2009.

58 Guy N. Rothblum, Salil Vadhan, and Avi Wigderson. Interactive proofs of proximity: Delegating computation in sublinear time. In *Proceedings of the 45th annual ACM Symposium on Theory of Computing (STOC)*, 2013.

59 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996. doi:10.1137/S0097539793255151.

60 Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992.

61 Alexander A Sherstov. The multiparty communication complexity of set disjointness. In *Proceedings of the 44th symposium on Theory of Computing*, pages 525–548. ACM, 2012.

62 R. Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, STOC '87, pages 77–82, New York, NY, USA, 1987. ACM. doi:10.1145/28395.28404.

63 Madhu Sudan. *Efficient Checking of Polynomials and Proofs and the Hardness of Approximation Problems*. PhD thesis, University of California at Berkeley, Berkeley, CA, USA, 1992. UMI Order No. GAX93-30747.

64 Madhu Sudan. *Efficient Checking of Polynomials and Proofs anf the Hardness of Approximation Problems*, volume 1001 of *Lecture Notes in Computer Science*. Springer, 1995. doi:10.1007/3-540-60615-7.

65 Justin Thaler. Semi-streaming algorithms for annotated graph streams. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 17:1–17:14, 2016. doi:10.4230/LIPIcs.ICALP.2016.17.

66 Salil P. Vadhan. On transformation of interactive proofs that preserve the prover's complexity. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 200–207, 2000. doi:10.1145/335305.335330.

67 Leslie G. Valiant. Short monotone formulae for the majority function. *J. Algorithms*, 5(3):363–366, 1984. doi:10.1016/0196-6774(84)90016-6.

68 Emanuele Viola. Guest column: correlation bounds for polynomials over {0 1}. *SIGACT News*, 40(1):27–44, 2009. doi:10.1145/1515698.1515709.

69 Richard Ryan Williams. Strong ETH breaks with Merlin and Arthur: Short non-interactive proofs of batch evaluation. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 2:1–2:17, 2016. doi:10.4230/LIPIcs.CCC.2016.2.

A Talmudic Discussions

A.1 Why GKR and not other Interactive Proofs?

One may wonder whether we could base our upper bound on other interactive proofs from the literature. Other than the protocols of [37, 45], two other general purpose interactive proof-systems that come to mind are Shamir's[27] [60] protocol for IP = PSPACE and a recent protocol of Reingold, Rothblum and Rothblum [56] that gives constant-round interactive proofs for bounded-space computations.

[27] Indeed, here we specifically refer to Shamir's [60] protocol and not to the [52] protocol (on which [60] builds).

Shamir's protocol is not suitable for our needs both because it is not constant-round, and, perhaps more fundamentally, because the verifier in Shamir's protocol needs to access the low-degree extension of the input over a field that is only determined during the interaction (recall that the verifier in Shamir's protocol chooses a *random* prime p, and the players both work over the field of integers modulo p). For our purposes the field has to be fixed a priori (since we want the input for the IPP to be encoded under the LDE code corresponding to that field).

As for the protocol of [56], the latter does actually yield a constant-round HIP for $\mathcal{L}_{\mathsf{MOD3}}$ (which can be modified to yield an IPP for Enc-MOD3 as above) but the tradeoff that it offers between rounds and the verifier's complexity is exponentially worse than what we obtain. More specifically, for every constant $r \geq 1$, the [56] protocol yields a $2^{\tilde{O}(r)}$-round HIP for $\mathcal{L}_{\mathsf{MOD3}}$ with verification time roughly $2^{\tilde{O}(r)} \cdot k^{1/r}$. In contrast, we obtain an $O(r^2)$-round HIP with verification time roughly $\mathsf{poly}(r) \cdot k^{1/r}$.

A.2 An Alternative Candidate Language for the Round Hierarchy Theorem

The language for which we proved our round hierarchy consists of encodings of strings whose Hamming weight is divisible by 3. As described next, it seems as though a similar result can be obtained for a related language Enc-Maj that consists of encodings of strings $x \in \{0,1\}^k$ with $\mathsf{wt}(x) \geq k/2$, although there are some technical difficulties to overcome.

First note that the lower bound for Enc-Maj follows along the same lines as our lower bound for Enc-MOD3, where now we use the fact that $\mathsf{AC}_0[2]$ circuits cannot approximate the majority function [55, 62]. In contrast, showing an upper bound (i.e., an IPP or HIP) introduces some new difficulties. As explained in Section 1 (and formalized in Section 4), our upper bound for Enc-MOD3 is based on the observation that computing the sum, modulo 3, of the bits of an input string can be done by a (highly uniform) NC^1 circuit. Given this observation, we based our protocol on a variant of the GKR interactive proof for small-depth computations.

For Enc-Maj, we could similarly hope to base our protocol on an NC^1 circuit, but this time we need a circuit that computes the majority function. Obtaining such a circuit is less trivial and here we encounter some difficulties:

- Valiant [67] (see the presentation of Goldreich [25]) gave a *non-uniform* construction of an NC^1 circuit for majority. We could base our protocol on this result and obtain a *non-uniform* verifier (and in particular, its running time would be super-linear, although it would still have the desired query and communication complexities).
- The aforementioned construction of [67] can actually be shown to produce a (highly-uniform) *randomized* construction. That is, there exists a randomized logspace Turing machine that given as input 1^n, with all but exponentially vanishing probability, produces an NC^1 circuit, on n-bit strings, that computes the majority function correctly (on all inputs). We could have our verifier run this procedure to obtain the desired NC^1 circuit, but this would introduce an (exponentially small) completeness error, which we would like to avoid.
- Lastly, we mention that the celebrated [3] sorting network of Ajtai, Komlós and Szemerédi gives rise to a uniform (and deterministic) construction of an NC^1 circuit for majority (by sorting the input bits and outputting the median). This construction is quite complex and in particular we have not verifed whether it satisfies the uniformity condition that is required for the [45] result.[28]

[28] We note that other *partial*, but arguably simpler, de-randomization results of Valiant's formula have

B From HIPs to IPPs (Proof of Proposition 11)

The two main ingredients that we shall use to prove Proposition 11 are the well-known low (individual) degree test[29] for multivariate polynomials [59, 64, 4], and the self-correction procedure for polynomials [23, 64].

▶ **Lemma 29** (Individual Degree Test). *Let $d, m \in \mathbb{N}$ such that $dm < |\mathbb{F}|/10$ and $\varepsilon \in (0, 1/10)$. Denote by $\mathsf{Poly}_{d,m,\mathbb{F}}$ the set of all m-variate, individual degree d polynomials over $\mathbb{F}$. Then, there exists an ε-tester for $\mathsf{Poly}_{d,m,\mathbb{F}}$ with query complexity $dm \cdot \mathsf{poly}(1/\varepsilon)$.*

▶ **Lemma 30.** *Let $\varepsilon < 1/3$ and $d, m \in \mathbb{N}$ such that $d \leq |\mathbb{F}|$. There exists an algorithm (corrector) that, given $x \in \mathbb{F}^m$ and oracle access to an m-variate function $f : \mathbb{F}^m \to \mathbb{F}$ that is ε-close to a polynomial p of individual degree d, makes $O(d \cdot m)$ queries and outputs $p(x)$ with probability $9/10$. Furthermore, if f has total degree d, the algorithm outputs $p(x)$ with probability 1.*

Given Lemmas 29 to 30, we can now describe the IPP (with respect to some proximity parameter ε) for $\mathsf{LDE}_{\mathbb{F},H,m}(\mathcal{L})$. Recall that the verifier is given oracle access to a function $f : \mathbb{F}^m \to \mathbb{F}$ and the prover is given direct access to f. Assume, without loss of generality, that the HIP for $\mathcal{L}$ has soundness error $1/10$.[30]

First, the verifier and prover run the HIP protocol for $\mathcal{L}$ with respect to the input $f|_{H^m}$. (Recall that an HIP does not even query its input and therefore, so far, no queries have been made.) If the HIP verifier rejects then we immediately reject. Otherwise, the verifier outputs a pair $(z, \nu) \in \mathbb{F}^m \times \mathbb{F}$ (with the associated claim that $f(z) = \nu$). Then, the verifier runs the individual degree tester of Lemma 29 on f, with respect to proximity parameter ε, individual degree $|H| - 1$ (and soundness error $1/3$). If the low degree test rejects, the verifier immediately rejects. Lastly, the verifier decodes f at point z, using the self-correction procedure of Lemma 30, again with soundness error $1/10$. The procedure outputs a value ν'. The verifier accepts if $\nu = \nu'$ and otherwise it rejects.

Completeness follows from the perfect completeness of the HIP, the low degree test and the local self-correction. For soundness, let $f : \mathbb{F}^m \to \mathbb{F}$ be a function such that f is ε-far from $\mathsf{LDE}_{\mathbb{F},H,m}(\mathcal{L})$ and fix a cheating prover strategy $\mathcal{P}^*$. Consider first the case that f is ε-far from an individual degree $|H| - 1$ polynomial. In this case, by the low degree test, with probability at least $2/3$, the verifier rejects and we are done. Thus, we can assume that f is ε-close to some individual degree $|H| - 1$ polynomial $P : \mathbb{F}^m \to \mathbb{F}$. Observe that since f is ε-far from $\mathsf{LDE}_{\mathbb{F},H,m}(\mathcal{L})$ it must be the case that $P|_{H^m} \notin \mathcal{L}$.

We view the HIP as being applied to $P|_{H^m}$. By the soundness of the HIP, when the verifier interacts with any cheating prover (and in particular $\mathcal{P}^*$) with probability $9/10$ it either rejects (in which case we also reject) or it outputs a pair $(z, \nu) \in \mathbb{F}^m \times \mathbb{F}$ such that $P(z) \neq \nu$. The verifier reads the point z with self-correction and so, with probability at least $9/10$ it will obtain the actual value $\nu' = P(z)$ and reject when comparing ν' and ν. Thus, with probability $0.9^2 \geq 2/3$ our verifier rejects.

been obtained by [44] and [15]. However, these partial derandomizations do not seem to suffice for our purposes.

[29] Actually, the cited works provide a test for *total* degree. A test for *individual* degree (which is implicit in [35, Section 5.4.2]) can be obtained via a simple reduction (see, e.g., [43, Theorem A.8]).

[30] Indeed, parallel repetition of IPPs decreases their soundness error at an exponential rate (see [31, Appendix A] for details).

C The Sumcheck Protocol (Proof of Lemma 10)

In this appendix we prove Lemma 10.

We use a variant of the sumcheck protocol that takes r rounds, where for simplicity we assume that r divides m. We maintain the invariant that before the i^{th} rounds begins, both the verifier and the prover agree on values $w_1, \ldots, w_{i-1} \in \mathbb{F}^{m/r}$ and $\nu_{i-1} \in \mathbb{F}$, where $\nu_0 \stackrel{\text{def}}{=} 0$. For every $i \in [r]$, the i^{th} round of the sumcheck protocol is as follows.

1. The prover sends to the verifier the individual degree $|H| - 1$ polynomial $P_i : \mathbb{F}^{m/r} \to \mathbb{F}$ (by specifying its coefficients), defined as:

$$P_i(z) \stackrel{\text{def}}{=} \sum_{x_{i+1},\ldots,x_r \in H^{m/r}} P(w_1, \ldots, w_{i-1}, z, x_{i+1}, \ldots, x_r).$$

2. The verifier receives a polynomial $Q_i : \mathbb{F}^{m/r} \to \mathbb{F}$ (which is allegedly equal to P_i) and checks that $\sum_{z \in H^{m/r}} Q_i(z) = \nu_{i-1}$.
3. The verifier select uniformly at random $w_i \in \mathbb{F}^{m/r}$ and sends w_i to the prover.
4. Set $\nu_i \stackrel{\text{def}}{=} Q_i(w_i)$.

At the end of the protocol, the verifier outputs $((w_1, \ldots, w_r), \nu_r) \in \mathbb{F}^m \times \mathbb{F}$.

The running times and communication complexity of the protocol can be readily verified. We proceed to show that completeness and soundness hold.

C.1 Completeness

Let $P : \mathbb{F}^m \to \mathbb{F}$ be an individual degree $|H| - 1$ polynomial such that $\sum_{x \in H^m} P(x) = 0$. In this case, at every round $i \in [\rho]$, the prover sends the polynomial $Q_i \equiv P_i$. Hence, for every $i \in [r]$:

$$\begin{aligned}
\sum_{z \in H^{m/r}} Q_i(z) &= \sum_{z \in H^{m/r}} P_i(z) \\
&= \sum_{z \in H^{m/r}} \sum_{x_{i+1},\ldots,x_r \in H^{m/r}} P(w_1, \ldots, w_{i-1}, z, x_{i+1}, \ldots, x_r) \\
&= P_{i-1}(w_{i-1}) \\
&= Q_{i-1}(w_{i-1}) \\
&= \nu_{i-1}
\end{aligned}$$

and so all of the verifier's checks pass. At the end of the protocol the verifier outputs $((w_1, \ldots, w_r), \nu_r) \in \mathbb{F}^m \times \mathbb{F}$ and $\nu_r = P_r(w_r) = P(w_1, \ldots, w_r))$ as required.

C.2 Soundness

Let $P : \mathbb{F}^m \to \mathbb{F}$ be an individual degree $|H| - 1$ polynomial such that $\sum_{x \in H^m} P(x) \neq 0$ and fix a cheating prover strategy $\mathcal{P}^*$.

The next two claims relate the polynomials Q_i sent by the prover to the corresponding polynomials P_i (recall that P_i was defined as $P_i(z) = \sum_{x_{i+1},\ldots,x_r \in H^{m/r}} P(w_1, \ldots, w_{i-1}, z, x_{i+1}, \ldots, x_r)$). Recall that both polynomials depend only on $w_1, \ldots, w_{i-1}$.

▶ **Claim 31.** *If $Q_1 \equiv P_1$, then the verifier rejects with probability* 1.

Proof. Observe that $\sum_{x_1 \in H^{m/r}} P_1(x_1) = \sum_{z \in H^m} P(z) \neq 0$, and so, if $Q_1 \equiv P_1$, then the verifier rejects when testing that $\sum_{z \in H^{m/r}} Q_1(z) = \nu_0 = 0$. ◀

▶ **Claim 32.** *For every $i \in [r-1]$ and every $w_1, \ldots, w_{i-1} \in \mathbb{F}^{m/r}$, if $Q_i \not\equiv P_i$ then, with probability $1 - \frac{(m/r)\cdot|H|}{|\mathbb{F}|}$ over the choice of w_i, if $Q_{i+1} \equiv P_{i+1}$ then the verifier rejects.*

Proof. Since the (total degree $(m/r)\cdot(|H|-1)$) polynomials Q_i and P_i differ, by the Shwartz-Zippel lemma (Lemma 5), with probability $1 - \frac{(m/r)\cdot|H|}{|\mathbb{F}|}$ over the choice of $w_i \in_R \mathbb{F}^{m/r}$, it holds that $Q_i(w_i) \neq P_i(w_i)$. If the latter event occurs and the prover sends $Q_{i+1} \equiv P_{i+1}$, then the verifier rejects when testing whether $\sum_{z\in H^{m/r}} Q_{i+1}(z) = \nu_i$, since

$$\nu_i = Q_i(w_i) \neq P_i(w_i) = \sum_{z\in H^{m/r}} P_{i+1}(z) = \sum_{z\in H^{m/r}} Q_{i+1}(z). \qquad \blacktriangleleft$$

By Claims 31 and 32 and an application of the union bound, with probability $1-(r-1)\cdot\frac{(m/r)\cdot|H}{|\mathbb{F}|}$, if there exists an $i \in [r-1]$ such that $Q_i \not\equiv P_i$ but $Q_{i+1} \equiv P_{i+1}$ then the verifier rejects. However, by Claim 31, we can assume that $Q_1 \not\equiv P_1$ and so we get that with probability $1-(r-1)\cdot\frac{(m/r)\cdot|H}{|\mathbb{F}|}$ either the verifier rejects or $Q_r \not\equiv P_r$. Note that if $Q_r \not\equiv P_r$ then by the Shwartz Zippel Lemma with probability $1 - \frac{(m/r)\cdot|H|}{|\mathbb{F}|}$ it holds that $Q_r(w_r) \neq P_r(w_r)$ and therefore:

$$\nu_r = Q_r(w_r) \neq P_r(w_r) = P(w_1, \ldots, w_r)$$

and so the soundness condition holds, with soundness error $(r-1)\cdot\frac{(m/r)\cdot|H}{|\mathbb{F}|} + \frac{(m/r)\cdot|H}{|\mathbb{F}|} = \frac{m\cdot|H}{|\mathbb{F}|}$.

D Interactive Proof for Vanishing-Subcube (Proof of Proposition 14)

Let $\mathbb{F}$ be a constructible field ensemble, let $H \subseteq G \subseteq \mathbb{F}$ be ensembles of subsets, and let $m \in \mathbb{N}$. Recall that $\mathsf{Vanishing\text{-}Subcube}_{\mathbb{F},H,m,G}$ is the set of all functions $f : G^m \to \mathbb{F}$ that vanish on H^m (i.e., $f|_{H^m} \equiv 0$). We show that for every $r \in [m]$, there exists an $r+2$-round (public-coin) HIP for $\mathsf{Vanishing\text{-}Subcube}_{\mathbb{F},H,m,G}$, with respect to the code $\mathsf{LDE}_{\mathbb{F},G,m}$.

Recall that in an HIP with respect to the code $\mathsf{LDE}_{\mathbb{F},G,m}$, the input should be thought of as an m-variate polynomial P with individual degree $|G|-1$. The prover has direct access to P and the verifier needs to output a pair $(z,\nu) \in \mathbb{F}^m \times \mathbb{F}$, with the associated claim that $P(z) = \nu$.

For a given function $P : \mathbb{F}^m \to \mathbb{F}$, we define the polynomial $\tilde{P}(x) = \sum_{z\in H^m} \delta(z,x)\cdot P(z)$, where $\delta : \mathbb{F}^m \times \mathbb{F}^m \to \mathbb{F}$ is an individual degree $|H|-1$ polynomial such that for every $a, b \in H^m$, it holds that $\delta(a,b) = 1$ if $a = b$ and $\delta(a,b) = 0$ otherwise (and δ is arbitrary in $\mathbb{F}^{2m}\setminus H^{2m}$).[31]

To check that P is identically 0 in H^m, the verifier first chooses at random $r \in \mathbb{F}^m$ and sends r to the prover. Now, the prover and verifier run an interactive proof to check that $\tilde{P}(r) = 0$, by invoking the sumcheck protocol with respect to the summation $\sum_{z\in H^m} \delta(z,r)\cdot P(z) = 0$, where we observe that the polynomial $\delta(\cdot,r)\cdot P(\cdot)$ has individual degree $|H|+|G|-1$. If the sumcheck verifier rejects, then we immediately reject. Otherwise, the sumcheck verifier outputs a pair $(z,\nu) \in \mathbb{F}^m \times \mathbb{F}$, and the prover then sends the value $\nu' = P(z)$. Finally, the verifier checks that $\delta(z,r)\cdot\nu' = \nu$ and if so outputs (z,ν').

For completeness, note that if P is identically 0 in H^m, then $\tilde{P}$ is identically 0 in $\mathbb{F}^m$. In particular, with probability 1 over the choice of r it holds that $\tilde{P}(r) = \sum_{z\in H^m} \delta(z,r)\cdot P(h) = 0$. Thus, by the completeness of the sumcheck protocol, the sumcheck verifier outputs a pair

[31] We note that $\tilde{P}$ is in fact the low degree extension of the function P, when the latter is restricted to H^m.

(z, ν) such that $\delta(z, r) \cdot P(z) = 0$. The prover now sends the value $\nu' = P(z)$, and so the verifier's check that $\delta(z, r) \cdot \nu' = \nu$ passes, and it outputs the claim (z, ν'), which is correct since $P(z) = \nu'$.

As for soundness, if P is not identically 0 in H^m, then by definition, $\tilde{P}$ is not identically 0 in $\mathbb{F}^m$, and therefore by the Schwartz-Zippel lemma (see Lemma 5), with probability $1 - \frac{m \cdot (|H|-1)}{|\mathbb{F}|}$ over the choice of r, it holds that $\tilde{P}(r) \neq 0$. Thus, the sumcheck protocol is invoked on the sum $\sum_{z \in H^m} \delta(z, r) \cdot P(z) \neq 0$ and so, with probability $1 - \frac{m \cdot (|H|+|G|-2)}{|\mathbb{F}|}$ either the sumcheck verifier rejects, or it outputs a claim (z, ν) such that $\delta(z, r) \cdot P(z) \neq \nu$. Assuming the latter happens, if the prover now sends $\nu' = P(z)$, then the verifier rejects. Hence, it must send $\nu' \neq P(z)$, and so the verifier outputs the incorrect claim (z, ν').

E Efficiently Computing $\widetilde{\mathsf{MOD3}}_t$

Recall that $\widetilde{\mathsf{MOD3}}_t : \mathbb{K}^t \to \mathbb{K}$ was defined as the (unique) individual degree 2 polynomial such that for every $h \in \{0,1,2\}^t$ it holds that $\widetilde{\mathsf{MOD3}}_t(h) = \sum_{i \in [t]} h_i \pmod 3$. In this section we show that $\widetilde{\mathsf{MOD3}}$ is efficiently computable. Namely, that given a point $z \in \mathbb{K}^t$, one can compute $\widetilde{\mathsf{MOD3}}_t(z)$ in time $\mathsf{poly}(t, \log(|\mathbb{K}|))$.

▶ **Proposition 33.** *Let $\mathbb{K}$ be a constructible field ensemble. There exists a $\mathsf{poly}(t, \log(|\mathbb{K}|))$-time algorithm that given a point $z \in \mathbb{K}^t$ outputs the value $\widetilde{\mathsf{MOD3}}_t(z)$.*

Proof. To prove Proposition 33, we first show that for every $\sigma \in \{0,1,2\}$ and $i \in [t]$, we can construct a size $\mathsf{poly}(i)$ uniform arithmetic circuit over $\mathbb{K}$ that computes the function $F_i^{(\sigma)} : \mathbb{K}^i \to \mathbb{K}$, which is defined as the unique individual degree 2 polynomial such that:

$$\forall h \in \{0,1,2\}^i, \quad F_i^{(\sigma)}(h) = \begin{cases} 1 & \text{if } \sum_{i \in [t]} h_i = \sigma \pmod 3 \\ 0 & \text{otherwise} \end{cases}.$$

where the summation is over integers modulo 3. Despite their similarity, note that $\widetilde{\mathsf{MOD3}}_t$ is the low degree extension of a function that *computes* the sum modulo 3 of its input, whereas $F_t^{(\sigma)}$ is the low degree extension of a function that *indicates* whether the sum modulo 3 is congruent to σ.

Given arithmetic circuits that compute $\mathbb{F}_t^{(\sigma)}$, we can now compute $\widetilde{\mathsf{MOD3}}_t : \mathbb{K}^t \to \mathbb{K}$ as:

$$\widetilde{\mathsf{MOD3}}_t(z) = \sum_{\sigma \in \{0,1,2\}} \sigma \cdot F_i^{(\sigma)}(z), \tag{10}$$

where here the arithmetic is over the field $\mathbb{K}$, and the equality follows from the fact that both sides of the equation are polynomials of individual degree 2 that agree on $\{0,1,2\}^t$ and therefore must agree on $\mathbb{K}^t$. Thus, it remains to prove the following claim.

▶ **Claim 34.** *For every $\sigma \in \{0,1,2\}$ and $i \in \mathbb{N}$, there exists an arithmetic circuit of size $O\left(i^{\log_2(6)}\right)$ over $\mathbb{K}$ that computes $F_i^{(\sigma)}$.*

Proof. We prove the proposition for i's that are powers of two and note that the general case follows easily (e.g., by using a circuit of size that is the nearest power of two and fixing some of its inputs to 0).

The proof is by induction on i, where the base case $i = 1$, is trivial. Fix i (that is a power of two) and suppose that we have constructed arithmetic circuits for computing $F_i^{(\sigma)}$ for every $\sigma \in \{0,1,2\}$.

Fix $\tau \in \{0,1,2\}$. The main observation is that for every $z_1, z_2 \in \mathbb{K}^i$ it holds that

$$F_{2i}^{(\tau)}(z_1, z_2) = \sum_{\sigma \in \{0,1,2\}} F_i^{(\sigma)}(z_1) \cdot F_i^{(\tau - \sigma \bmod 3)}(z_2), \tag{11}$$

where the equality follows from the fact that that both sides of the equation are polynomials of individual degree 2 that agree on $\{0,1,2\}^i$ and therefore must agree on $\mathbb{K}^{2i}$.

Denoting by S_i the size of the arithmetic circuit that Equation (11) yields for $F_i^{(\sigma)}$, it holds that:

$$S_{2i} = 6 \cdot S_i + c = \cdots = 6^{\log(2i)} \cdot S_1 + c \cdot \sum_{j=0}^{i-1} 6^j = O\left((2i)^{\log_2(6)}\right),$$

where $c \leq 10$ is the constant overhead that arises from Equation (11). This concludes the proof of Claim 34. ◀

Proposition 33 now follows by combining Equation (10) and Claim 34. ◀

Cube vs. Cube Low Degree Test

Amey Bhangale[1], Irit Dinur[2], and Inbal Livni Navon[3]

1 **Department of Computer Science, Rutgers University, Piscataway, USA**
amey.bhangale@rutgers.edu
2 **Faculty of Computer Science and Mathematics, Weizmann Institute, Rehovot, Israel**
irit.dinur@weizmann.ac.il
3 **Computer Science and Mathematics, Weizmann Institute, Rehovot, Israel**
inbal.livni@weizmann.ac.il

Abstract

We revisit the Raz-Safra plane-vs.-plane test and study the closely related cube vs. cube test. In this test the tester has access to a "cubes table" which assigns to every cube a low degree polynomial. The tester randomly selects two cubes (affine sub-spaces of dimension 3) that intersect on a point $x \in \mathbb{F}^m$, and checks that the assignments to the cubes agree with each other on the point x. Our main result is a new combinatorial proof for a low degree test that comes closer to the soundness limit, as it works for all $\epsilon \geq \text{poly}(d)/|\mathbb{F}|^{1/2}$, where d is the degree. This should be compared to the previously best soundness value of $\epsilon \geq \text{poly}(m,d)/|\mathbb{F}|^{1/8}$. Our soundness limit improves upon the dependence on the field size and does not depend on the dimension of the ambient space.

Our proof is combinatorial and direct: unlike the Raz-Safra proof, it proceeds in one shot and does not require induction on the dimension of the ambient space. The ideas in our proof come from works on direct product testing which are even simpler in the current setting thanks to the low degree.

Along the way we also prove a somewhat surprising fact about connection between different agreement tests: it does not matter if the tester chooses the cubes to intersect on points or on lines: for every given table, its success probability in either test is nearly the same.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes

Keywords and phrases Low Degree Test, Probabilistically Checkable Proofs, Locally Testable Codes

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.40

1 Introduction

Low degree tests are local tests for the property of being a low degree function. These were the first property testing results that were discovered, and are an important component in PCP constructions. Such tests were studied in the 1990's and their ballpark soundness behavior was more or less understood. In this work we revisit these tests and give a new and arguably simpler analysis for the cube vs. cube low degree test. Our proof method allows us to get a soundness guarantee that is much closer to the conjectured optimal value. Discovering the precise point in which soundness starts to hold is an intriguing open question that captures an interesting aspect of local-testing in the small soundness regime.

Let us begin with a short introduction to low degree tests. A low degree test can be described as a game between a prover and a verifier, in which the prover wants to convince the verifier that a function $f : \mathbb{F}^m \to \mathbb{F}$ is a low degree polynomial. The most straightforward


© Amey Bhangale, Irit Dinur, and Inbal Rachel Livni Navon;
licensed under Creative Commons License CC-BY

way for the prover to specify f would be to give its value on each point $x \in \mathbb{F}^m$. However, in this way, to check that f has degree at most d the verifier would have to read f on at least $d+2$ points. If we want a verifier that makes fewer queries while keeping the error small, it is useful to move to a more redundant representation of f. For example, the verifier can ask the prover to specify for every cube (affine subspace of dimension 3) $C \subset \mathbb{F}^m$, a function $f_C : C \to \mathbb{F}$ that is defined on the cube and is obtained by restricting f to that cube. This is called a "cubes-table", and similarly one can consider a lines table (with an entry for every line), or a planes table (with an entry for each plane).

Thus, in the cubes representation of a low degree function $f : \mathbb{F}^m \to \mathbb{F}$, we have a table entry $T(C)$ for every cube C and the value of that entry is supposed to be $T(C) = f|_C$. A general cubes table is a table $T(\cdot)$ indexed by all possible cubes and the C-th entry is a low degree function on the cube C. Each $T(C)$ is viewed as a local function. Indeed the number of bits needed to specify $T(C)$ is only $O(d^3 \log |\mathbb{F}|)$ which is much smaller than $\binom{m+d}{d} \log |\mathbb{F}|$ - the number of bits needed to represent a general degree d function f on $\mathbb{F}^m$.

The prover may cheat, as provers do, by giving a cubes table whose entries cannot be "glued together" into any one global low degree function. This is where the *agreement* test comes in. The verifier can check the table by reading two entries corresponding to two cubes that have a non-trivial intersection, and checking that the function $T(C_1)$ and the function $T(C_2)$ agree on points in the intersection of $C_1 \cap C_2$.

Test 1 Cube vs. Cube agreement test.

1. Select a point $x \in \mathbb{F}^m$.
2. Pick affine cubes C_1, C_2 randomly conditioned on $C_1, C_2 \ni x$.
3. Read $T(C_1), T(C_2)$ from the table and accept iff $T(C_1)(x) = T(C_2)(x)$.

Let $\alpha_{\mathcal{C}x\mathcal{C}}(T)$ be the *agreement* of the table T, i.e. the probability of acceptance of the test.

The test is local in that it accesses only two cubes. Different tests may differ in the distribution underlying the agreement test (for example, Raz and Safra look at two planes that intersect in a line, which clearly is a different distribution from choosing two planes that intersect in a point), but they all check agreement on the intersection, so we generally refer to all of these as agreement tests.

The interesting point, as proven by both Raz and Safra in [11], and by Arora and Sudan in [1], is that such tests have small soundness error. For example, the plane vs. plane theorem of Raz Safra is as follows,

▶ **Theorem 1** (Raz-Safra [11]). *There is some $\delta > 0$ such that for every d and prime power q and every $m \geq 3$ the following holds. Let $\mathbb{F}$ be a finite field $|\mathbb{F}| = q$, and let $T(\cdot)$ be a planes table, assigning to each plane $P \subset \mathbb{F}^m$ a bivariate degree d polynomial $T(P) : P \to \mathbb{F}$. Let $\alpha_{\mathcal{P}\ell\mathcal{P}}(T)$ be as defined in Test 2.*

For every $\epsilon \geq (md/q)^\delta$, if $\alpha_{P\ell P}(T) \geq \epsilon$ then there is a degree d function $g : \mathbb{F}^m \to \mathbb{F}$ such that $T(P) = g|_P$ on an $\Omega(\epsilon)$ fraction of the planes.

A similar theorem was proven by Arora and Sudan for T a lines table and for a natural test that checks if two intersecting lines agree on the point of intersection.

These results are called low degree tests although it makes sense to think of them as theorems relating local agreement to global agreement. We refer to them as low degree *agreement* test theorems.

Test 2 The Raz-Safra Plane vs. Plane agreement test.

1. Select an affine line $\ell \subset \mathbb{F}^m$.
2. Choose affine planes P_1, P_2 randomly conditioned on $P_1, P_2 \supset \ell$.
3. Read $T(P_1), T(P_2)$ from the table and accept iff $T(P_1)(x) = T(P_2)(x)$ for all $x \in \ell$.

Let $\alpha_{\mathcal{P}\ell\mathcal{P}}(T)$ be the *agreement* of the table T, i.e. the probability of acceptance of the test.

1.1 Towards the soundness threshold

The most important aspect of the low degree agreement theorems of [11, 1] is the fact that they have small soundness. Small soundness means that a cheating prover won't be able to fool the verifier into accepting with even a tiny $\epsilon > 0$ probability, unless the table has some non-trivial agreement with a global low degree function. Small soundness of low degree tests was used inside PCP constructions for getting PCPs with the smallest known soundness error. The fact that soundness holds for all values of $\epsilon \geq (d/q)^\delta$ was sufficient for the PCP constructions of [11, 1]. It is likely that finding the minimal threshold beyond which soundness is guaranteed to hold will be important for determining the best possible PCP gaps.

Regardless of the PCP application, this encoding of a function f by its restrictions to cubes (or to planes) is quite natural, and is a rare example of a property that has such strong testability. The low degree agreement test theorems guarantee that even the passing of the test with tiny ϵ probability has non-trivial structural consequences. Perhaps the best known comparable scenario is that of the long code, defined in [2], that has similar properties, and for which an extensive line of work has been able to determine the precise threshold of soundness. Another setting with a similarly strong soundness is related to the inverse theorems for the Gowers uniformity norms. In that setting the function is given as a points-table, and the Gowers norm measures success in a low degree test, so it is not altogether dissimilar from the situation here.

To summarize, one of our goals is to pinpoint the absolute minimal soundness value for which a theorem as above holds. Can this threshold be, as it is in the aforementioned cases, as small as the value of a random assignment? In other words, could it be true that for every table whose agreement parameter is an additive $\epsilon > 0$ above the value that we expect from a random table, already some structure exists?

The best known value for δ for the plane vs. plane test is due to Moshkovitz and Raz who proved in [10] that the plane vs. plane test has soundness for all $\epsilon \geq \text{poly}(d)/q^{1/8}$. But what is the correct exponent of q ?

We make progress on this question not for the plane vs. plane test but rather for the cube vs. cube test. For our test, since the intersection consists of one point, the soundness can not go below $1/q$ because the agreement of *every* table, even a random one, is always at least $1/q$.

Our main theorem is,

▶ **Theorem 2.** *There exist constants $\beta_1, \beta_2 > 0$ such that for every d, large enough prime power q and every $m \geq 3$ the following holds:*

Let $\mathbb{F}$ be a finite field, $|\mathbb{F}| = q$. Let T be a cubes table, assigning to each cube $C \subset \mathbb{F}^m$ a degree d polynomial $T(C) : C \to \mathbb{F}$. Let $\alpha_{\mathcal{C}x\mathcal{C}}(T)$ be as defined in Test 1. If $\alpha_{\mathcal{C}x\mathcal{C}}(T) \geq \epsilon$ for $\epsilon \geq \beta_1 d^4/q^{1/2}$, then there is a degree d function $g : \mathbb{F}^m \to \mathbb{F}$ such that $T(C) = g|_C$ on an $\beta_2\epsilon$ fraction of the cubes.

The improvement over previous theorems is that the dependence on q is $1/q^{1/2}$ compared to $1/q^{1/8}$. It is an intriguing question whether the dependence on q can be made inversely linear, i.e. $1/q$.

▶ Remark. We don't know the precise dependence of ϵ on the degree d. In this work we made no attempt to optimize this dependence. We would like to point out that our proof can be modified to change the dependence from d^4 to d^3. See Remark 3.3 for more details.

1.2 Simplified analysis

While the line vs. line test considered by Arora and Sudan [1] is the most natural to come up with, it is rather difficult to analyze. In contrast, one of the captivating aspects of the Raz-Safra proof is that it is combinatorial, and the low degree aspect of the table plays a role only in that it guarantees distance between distinct polynomials on a line. Our analysis continues this combinatorial approach, and further simplifies it. Unlike the Raz-Safra proof, we do not need to use induction on the dimension of the ambient space m but rather recover the global structure from T "in one shot". We rely on ideas from direct product testing, [5, 9, 7], and on some spectral properties of incidence graphs such as the cube-point graph.

1.3 Proof Outline

Given a table T, whose agreement is some small ϵ, the proof must somehow come up with the global low degree function $g : \mathbb{F}^m \to \mathbb{F}$ and then argue that on many of the cubes indeed $T(C) = g|_C$. Naively, we might try to define g at each point x according to the most common value among all cubes containing x. This is a viable approach when the agreement is close to 1, as is done, e.g. in the linearity testing theorem of [3]. However, when the agreement is a small $\epsilon > 0$, this will simply not work as we can see by considering the table half of whose entries are $T(C) \equiv 0$ and the other half $T(C) \equiv 1$. The agreement of this table is an impressive $\alpha_{\mathcal{C}x\mathcal{C}}(T) = 1/2$, and yet the suggested definition of g according to majority will yield a random function that might be quite far from any low degree function.

We get around this problem by taking a *conditional majority*. For every point $x \in \mathbb{F}^m$ and value $\sigma \in \mathbb{F}$ we consider only cubes containing x for which $T(C)(x) = \sigma$. These cubes already agree with each other on x and are thus likely to agree on any other point of their intersection. Since the cubes containing x cover every $y \in \mathbb{F}^m$, we can define a function $f_{x,\sigma} : \mathbb{F}^m \to \mathbb{F}$ on the entire space $\mathbb{F}^m$ by taking the most popular value among these cubes (i.e. the set of cubes whose value on x is σ). We choose a best σ for each x and are left with a global function f_x for each x.

The proof proceeds in three steps.

- Local structure: We show that this conditional majority definition is good, obtaining for each x and σ a function $f_x : \mathbb{F}^m \to \mathbb{F}$ that is "local" in that it comes from the cubes containing a point x. This is done in Section 3.1.
- Global Structure: We then show that there are many pairs x, y for which $f_x \approx f_y$ thus finding a global g that agrees with many of the cubes. This is done in Section 3.2.
- Low Degree: Finally, we show that g is very close to a true low degree function. This is done by reduction to the Rubinfeld-Sudan low degree test [12] that works in the high-soundness regime. This is done in Section 3.3.

1.4 Agreement tests: low degree tests and direct product tests

The proof outline above resembles works on direct product testing, and this is no coincidence. The low degree testing setting can be generalized to a more abstract "agreement testing"

in which a function $f : X \to \Sigma$ is represented not as a truth table but as a collection of restrictions $(f|_S)_{S\in\mathcal{S}}$ where $\mathcal{S} = \{S \subset X\}$ is a collection of subsets of X. A natural agreement test can be defined and studied. This type of question was first suggested in work of Goldreich and Safra [8] in an attempt to separate the algebraic aspect of the low degree test from the combinatorial. There has been a follow-up line of work on this, [6, 5, 9, 7], focusing especially on the case where X is a finite set, $X = [n]$, and $\mathcal{S}$ is the collection of all k-element subsets of X.

In the work here we bring some of the ideas from that line of work, most notably from [9], back to the low degree testing question. The fact that our table entries have low degree gives us extra power which makes our proof simpler than that in the abstract setting, yielding a particularly direct proof of a low degree agreement test.

Our proof makes an explicit use of the expansion properties of the relevant incidence graphs (cube vs. line, cube vs. point etc.). This allows us to prove that for every table T, different tests have similar agreement.

▶ **Lemma 3.** *Let T be a planes table, and let $\alpha_{\mathcal{P}x\mathcal{P}}(T)$ be the success probability of a test with two planes that intersects on a point. Let $\alpha_{\mathcal{P}\ell\mathcal{P}}(T)$ be the success probability of Test 2, then*

$$\alpha_{\mathcal{P}x\mathcal{P}}(T)\left(1-\frac{d}{q}\right) \leq \alpha_{\mathcal{P}\ell\mathcal{P}}(T) \leq \alpha_{\mathcal{P}x\mathcal{P}}(T) + \frac{1}{q}(1+o(1)).$$

In fact, we proved a more general equivalence between tests, the general statement appears on Section 4.

2 Preliminaries and Notations

2.1 Notations

All the graphs we discuss throughout the paper are bipartite bi-regular graphs. Given such graph G, whose sides are A, B we denote by $\mathbf{1}$ the all one vector, its size will be implied by the context. For a subset of vertices $A' \subset A$, we denote by $\mathbf{1}_{A'}$ the indicator vector for A'. For a vertex $a \in A$, we denote by $N(a) \subseteq B$ the neighbors of a in G.

We use normalized inner product, such that for $x, y \in \mathbb{R}^n$, $\langle x, y\rangle = \frac{1}{n}\sum_i x_i y_i$, which means that $\langle \mathbf{1}, \mathbf{1}\rangle = 1$. The norm is defined by $\|x\| = \sqrt{\langle x, x\rangle}$.

We use the notation $x \sim S$ to denote x being sampled uniformly at random (u.a.r) from the set S, in case this set S equals the entire space, we omit this symbol and simply write $\mathbf{Pr}_a$ or $\mathbf{E}_a$ to describe choosing a uniform vertex $a \in A$. We use the notation $\mathbb{I}(E)$ to denote the indicator random variable of the event E.

For two vectors u, v, we use the notation $u \overset{\gamma}{\approx} v$ if u and v are equal on at least $1-\gamma$ of the coordinates.

Fix a vector space $\mathbb{F}^m$. An affine space of S dimension k is defined by $k+1$ vectors $x_0, x_1, \ldots, x_k$ such that $x_1, \ldots, x_k$ are linearly independent,

$$S = x_0 + \mathtt{span}(x_1, \ldots, x_k) = \{x_0 + t_1 x_1 + \ldots t_k x_k \mid t_1, \ldots, t_k \in \mathbb{F}.\}$$

A *line* is a 1-dimensional affine space, a *plane* is a 2-dimensional affine space, and a *cube* is a 3-dimensional affine space. We will denote the set of all lines and cubes by $\mathcal{L}$ and $\mathcal{C}$ be respectively. For a point $x \in \mathbb{F}^m$ let

$$\mathcal{L}_x = \{\ell \in \mathcal{L} \mid \ell \ni x\} \qquad \mathcal{C}_x = \{C \in \mathcal{C} \mid C \ni x\}.$$

Similarly for a line $\ell \in \mathcal{L}$ let $\mathcal{C}_\ell$ be the set of all cubes that contains ℓ.

2.2 Spectral Expansion Properties

In this section, we prove two properties of bi-regular bipartite graphs with good spectral parameters. In an expander, the following is well known: if we sample a random neighbor of a small, but not too small, set of vertices, we get a nearly uniform distribution over the entire set of vertices. For our purposes, we will require something more. We need to consider not only the distribution over the vertices, but also the distribution over the edges. This is done in two lemmas below.

▶ **Definition 4.** Let $G = (A \cup B, E)$ be a bi-regular bipartite graph, and let $M \in \mathbb{R}^{A \times B}$ be the adjacency matrix normalized such that $\|M\mathbf{1}\| = 1$, denote by $\lambda(G)$ the value

$$\lambda(G) = \max_{v \perp \mathbf{1}} \left\{ \frac{\|Mv\|}{\|v\|} \right\}.$$

This is really the second largest singular value of M, with a different normalization (such that the maximal singular value equals 1).

▶ **Definition 5.** Let $G = (A \cup B, E)$ be a bi-regular bipartite graph and let $B' \subseteq B$ be a subset of vertices. Define the following two distributions $D_i : A \times B \cup \perp \to [0, 1]$ for $i = 1, 2$.

- D_1 : Pick $b \in B'$ u.a.r. then pick $a \in N(b)$ u.a.r.
- D_2 : Pick $a \in A$ u.a.r. If $B' \cap N(a) = \emptyset$, return $\perp$. Else, pick $b \in N(a) \cap B'$ u.a.r.

Clearly if $B' = B$ then $D_1 = D_2$. Moreover, if G is sufficiently expanding, then even for smaller $B' \subsetneq B$, the distributions are similar. Indeed, for any event defined on the edges, i.e. a subset $E' \subset E$, the following lemma shows that the probability of E' is roughly the same under the two distributions.

▶ **Lemma 6.** *Let D_1, D_2 as defined in Definition 5. Let $G = (A \cup B, E)$ be a bi-regular bipartite graph, then for every subset $B' \subset B$ of measure $\mu > 0$ and every $E' \subset E$*

$$\left| \Pr_{(a,b)\sim D_1}[(a, b) \in E'] - \Pr_{(a,b)\sim D_2}[(a, b) \in E'] \right| \leq \frac{\lambda(G)}{\sqrt{\mu}}.$$

Where it is understood that if D_2 output $\perp$, we treat it as if $(a, b) \notin E'$.

We now state a similar lemma, for sampling two adjacent edges instead of a single edge. We will need the graph to satisfy one more requirement.

▶ **Definition 7.** Let $G = (A \cup B, E)$ be a bi-regular bipartite graph, such that every two distinct $b_1, b_2 \in B$ have exactly the same number of common neighbors (i.e for all distinct $b_1, b_2 \in B$, $|N(b_1) \cap N(b_2)|$ is the same), and this number is non-zero. Let $B' \subseteq B$ be a subset of vertices, we define the following distributions $D_i : (A \times B \times B) \cup \perp \to [0, 1]$, for $i = 3, 4$.

- D_3 : Pick $b_1, b_2 \in B'$ u.a.r. then pick $a \in N(b_1) \cap N(b_2)$ u.a.r.
- D_4 : Pick $a \in A$ u.a.r. If $B' \cap N(a) = \emptyset$, return $\perp$. Else, pick $b_1, b_2 \in N(a) \cap B'$ u.a.r.

▶ **Lemma 8.** *Let D_3, D_4 be as defined in Definition 7. Let $G = (A \cup B, E)$ be a bi-regular bipartite graph, such that every two distinct $b_1, b_2 \in B$ have exactly the same number of common neighbors (i.e for all distinct $b_1, b_2 \in B$, $|N(b_1) \cap N(b_2)|$ is the same), and this number is non-zero. Then for every subset $B' \subset B$ of measure $\mu > 0$ and every $E' \subset E$*

$$\left| \Pr_{a,b_1,b_2\sim D_3}[(a, b_1)(a, b_2) \in E'] - \Pr_{a,b_1,b_2\sim D_4}[(a, b_1)(a, b_2) \in E'] \right| \leq \frac{2\lambda(G)}{\mu} + \frac{1}{\mu^2 d_A} + \frac{1}{\mu^2 |B|},$$

where d_A is the degree on A side, and it is understood that if D_4 output $\perp$, we treat it as if $(a, b) \notin E'$.

The proofs of these two lemmas appear in Appendix B.

2.3 Inclusion Graphs and Their Spectral Gap

We record here the expansion of several bi-partite *inclusion graphs* that will be relevant for our analysis. We prove the claims about these spectral gaps in Appendix A. Unless otherwise stated, $G(A, B)$ denotes a bipartite inclusion graph between A and B where $a \in A$ is connected to $b \in B$ if $a \subseteq b$. The relation of containment will be clear from the sets A and B.

For example, the in the graph $G_1(\mathcal{L} \setminus \mathcal{L}_x, \mathcal{C}_x)$, the left side vertices A are all the lines that do not contain $x \in \mathbb{F}^m$, and the right side vertices are all the cubes that contain x. There is an edge between a line ℓ and a cube C if $\ell \subset C$.

Recall Definition 4 of $\lambda(G)$ for a bipartite graph G.

▶ **Lemma 9.** *We have for every $m \geq 6$,*
(1) *For $G_1(\mathcal{L} \setminus \mathcal{L}_x, \mathcal{C}_x)$, $\lambda(G_1) \approx \frac{1}{\sqrt{q}}$.*
(2) *For $G_2(\mathcal{L}_x, \mathcal{C}_x))$, $\lambda(G_2) \approx \frac{1}{q}$.*
(3) *For $G_3(\mathbb{F}^m \setminus \ell, \mathcal{C}_\ell)$, $\lambda(G_3) \approx \frac{1}{\sqrt{q}}$.*
(4) *For $G_4(\mathbb{F}^m, \mathcal{C})$, $\lambda(G_4) \approx \frac{1}{q^{3/2}}$.*
(5) *For $G_5(\mathbb{F}^m \setminus \{x\}, \mathcal{C}_x)$, $\lambda(G_5) \approx \frac{1}{q}$.*
And for every $m \geq 3$
(6) *For $G_6(\mathbb{F}^m, \mathcal{L})$, $\lambda(G_6) \approx \frac{1}{\sqrt{q}}$.*
where $\approx$ denotes equality up to a multiplicative factor of $1 \pm o(1)$, and $o(1)$ denotes a function that approaches zero as $q \to \infty$.

In general one can see that $\lambda \approx \frac{1}{\sqrt{q^p}}$ where p is the number of degrees of freedom left after choosing a left hand vertex. We prove this lemma in Appendix A.

3 Proof of the Main Theorem

In this section we prove Theorem 2 in three steps - local structure, global structure and finally proving the agreement with a low degree polynomial. These parts are proved in the subsequent subsections.

Let T be a degree d cubes table, i.e. for every $C \in \mathcal{C}$, $T(C) : C \to \mathbb{F}$ is a degree d polynomial. Further assume that $\alpha_{\mathcal{C}x\mathcal{C}}(T) \geq \epsilon$, where $\epsilon = \Omega(d^4/\sqrt{q})$.

3.1 Local Structure

In this section we show that for many points $x \in \mathbb{F}^m$, there exists a function $f_x : \mathbb{F}^m \to \mathbb{F}$ for which $f_x|_C \overset{2\gamma}{\approx} T(C)$ for a good fraction of the cubes containing x, for $\gamma = \Omega(1/d^3)$. Recall that $\overset{2\gamma}{\approx}$ means that the two functions agree on $1 - 2\gamma$ fraction of the points in their domain.

For each $x \in \mathbb{F}^m$ and $\sigma \in \mathbb{F}$, we define

$$\mathcal{C}_{x,\sigma} = \{C \in \mathcal{C}_x | T(C)(x) = \sigma\}.$$

Following [9] we have the following important definition,

▶ **Definition 10** (Excellent pair)**.** (x, σ) is $(\frac{\epsilon}{2}, \gamma)$-excellent if:

1. $\mathbf{Pr}_{C \in \mathcal{C}_x}[C \in \mathcal{C}_{x,\sigma}] \geq \frac{\epsilon}{2}$.

2. Let C_1, ℓ, C_2 be chosen by the following probability distribution, $C_1 \in \mathcal{C}_{x,\sigma}$ u.a.r, $\ell \subset C_1$ a random line that contains x and $C_2 \in \mathcal{C}_{x,\sigma} \cap \mathcal{C}_\ell$ (a random cube in $\mathcal{C}_{x,\sigma}$ that contains ℓ).

$$\mathop{\mathbf{Pr}}_{C_1,\ell,C_2}[T(C_1)_{|\ell} \neq T(C_2)_{|\ell}] \leq \gamma.$$

A point $x \in \mathbb{F}^m$ is $(\frac{\epsilon}{2}, \gamma)$-excellent, if exists $\sigma \in \mathbb{F}$ such that (x, σ) is $(\frac{\epsilon}{2}, \gamma)$-excellent.

Note that in the definition of excellent, the marginal distribution of both C_1, C_2 is uniform in $\mathcal{C}_{x,\sigma}$.

In the sequel, we fix $\gamma = \Omega(1/d^3)$ and say that a point is excellent if it is $(\frac{\epsilon}{2}, \gamma)$-excellent. We now state the main lemma in this section.

▶ **Lemma 11** (Local Structure). *For $\gamma = \Omega(\frac{1}{d^3})$, let T be a cubes table that passes Test 1 with probability larger than $\epsilon = \Omega(\frac{d^4}{\sqrt{q}})$, then at least $\frac{\epsilon}{3}$ of the points $x \in \mathbb{F}^m$ are excellent, and for each excellent x there exist a function $f_x : \mathbb{F}^m \to \mathbb{F}$ such that*

$$\mathop{\mathbf{Pr}}_{C \sim \mathcal{C}_x}[T(C) \overset{2\gamma}{\approx} f_{x|_C}] \geq \frac{\epsilon}{4}.$$

We will consider the distribution $\mathcal{D}$ on (x, ℓ, C_1, C_2) obtained by choosing x uniformly, choosing $\ell \in \mathcal{L}_x$ uniformly, and then choosing $C_1, C_2 \in \mathcal{C}_\ell$ uniformly.

This distribution induces a distribution $(x, T(C_1)(x))$ on pairs of point x and value $\sigma \in \mathbb{F}$.

▶ **Claim 12.** *For every $\gamma = \Omega(\frac{1}{d^3})$,*

$$\mathop{\mathbf{Pr}}_{(x,\sigma)}[(x, \sigma) \textit{ is } (\tfrac{\epsilon}{2}, \gamma) \textit{ - excellent}] \geq \frac{\epsilon}{3}.$$

Proof. We consider (x, ℓ, C_1, C_2) chosen according to $\mathcal{D}$, and we note that the marginal distribution over all elements is uniform. We also write $\sigma = T(C_1)(x)$. We define the following events on (x, ℓ, C_1, C_2):

1. E : "ℓ is confusing for x": $T(C_1)(x) = T(C_2)(x)$, $T(C_1)_{|\ell} \neq T(C_2)_{|\ell}$.
2. H : "x, C_1 is heavy": $\mathbf{Pr}_{C \sim \mathcal{C}_x}[T(C)(x) = T(C_1)(x)] \geq \frac{\epsilon}{2}$

Since $T(C_1)_{|\ell}, T(C_2)_{|\ell}$ are two degree d polynomials, and x is a random point in ℓ,

$$\mathop{\mathbf{Pr}}_{(x,\ell,C_1,C_2)}[E] \leq \frac{d}{q}.$$

Using the fact that $\alpha_{\mathcal{C}x\mathcal{C}}(T) \geq \epsilon$, and averaging, we get

$$\mathop{\mathbf{Pr}}_{(x,\ell,C_1,C_2)}[H] \geq \frac{\epsilon}{2}. \tag{1}$$

Instead of picking C_1 as a uniform cube containing x, we can choose it by the following process, pick σ proportional to its weight in $\mathcal{C}_x$, then pick $C_1 \sim \mathcal{C}_{x,\sigma}$. This process describes the same distribution.

Note that after deciding x, σ, the event H is already determined, so (1) becomes $\mathbf{Pr}_{x,\sigma}[H] \geq \epsilon/2$. Also, notice that conditioned on x, σ, the distribution $\mathcal{D}$ is choosing C_1 uniformly from $\mathcal{C}_{x,\sigma}$ and then $\ell \subset C_1$ a random line containing x and then C_2 a random cube containing ℓ (and we do not require that $T(C_2)(x) = \sigma$). The event H is already fixed by x, σ, but the event E will occur only if $C_2 \in \mathcal{C}_{x,\sigma}$ and also $T(C_1)|_\ell \neq T(C_2)|_\ell$.

We want to bound the probability of x, σ such that $H = 1$, but $\mathbf{E}_{C_1,\ell,C_2}[E|x,\sigma] \leq \gamma \cdot \frac{\epsilon}{2}$. We know that

$$\mathop{\mathbf{E}}_{x,\sigma}[\mathbf{Pr}[H \wedge E \mid x,\sigma]] = \mathbf{Pr}[H \wedge E] \leq \mathbf{Pr}[E] \leq \frac{d}{q}.$$

Therefore, by averaging, the probability over x, σ that we have $\mathbf{Pr}[H \wedge E|x,\sigma] > \epsilon\gamma/2$ is at most $\frac{d/q}{\epsilon\gamma/2}$. So for at least $\epsilon/2 - \frac{d/q}{\epsilon\gamma/2} \geq \epsilon/3$ of the pairs x, σ, we have that both H occurs, and that $\mathbf{E}_{C_1,\ell,C_2}[E|x,\sigma] \leq \epsilon\gamma/2$.

We end by showing that such x, σ are excellent. The first requirement follows by the fact that H occurs, for the second we need to show that for $C_1 \in \mathcal{C}_{x,\sigma}$, a uniform $\ell \in \mathcal{C}_1$ and a uniform $C_2 \in \mathcal{C}_{x,\sigma} \cap C_\ell$ the probability of $T(C_1)_{|\ell} \neq T(C_2)_{|\ell}$ is lower than γ.

We notice that after fixing (x, σ), the distribution $\mathcal{D}$ chooses $C_1 \in \mathcal{C}_{x,\sigma}$, a uniform $\ell \in \mathcal{C}_1$, but then a uniform $C_2 \in \mathcal{C}_\ell$.

The event E can be written as $E = E_1 \wedge E_2$ where E_1 is the event "$T(C_1)(x) = T(C_2)(x)$" and E_2 is the event "$T(C_1)_{|\ell} \neq T(C_2)_{|\ell}$". In this notation

$$\begin{aligned}
\mathop{\mathbf{E}}_{C_1,\ell,C_2}[E|x,\sigma] &= \mathop{\mathbf{E}}_{C_1,\ell,C_2}[E_1 \wedge E_2|x,\sigma] \\
&= \mathop{\mathbf{E}}_{C_1,\ell,C_2}[E_1|x,\sigma] \mathop{\mathbf{E}}_{C_1,\ell,C_2}[E_2|E_1,x,\sigma] \\
&\geq \frac{\epsilon}{2} \cdot \mathop{\mathbf{E}}_{C_1,\ell,C_2}[E_2|E_1,x,\sigma]. && \text{(since } H \text{ occurs)}
\end{aligned}$$

We notice that if E_1 occurs, then $C_2 \in \mathcal{C}_{x,\sigma}$, therefore

$$\mathop{\mathbf{E}}_{C_1,\ell,C_2}[T(C_1)_{|\ell} \neq T(C_2)_{|\ell}|C_2 \in \mathcal{C}_{x,\sigma},x,\sigma] \leq \frac{2}{\epsilon} \cdot \mathop{\mathbf{E}}_{C_1,\ell,C_2}[E|x,\sigma] \leq \frac{2}{\epsilon}\frac{\epsilon}{2}\gamma \leq \gamma,$$

which means that (x, σ) is $(\frac{\epsilon}{2}, \gamma)$ - excellent. ◀

For each (x, σ) we define $f_{x,\sigma}$ by plurality over all cubes $C \in \mathcal{C}_{x,\sigma}$.

▶ **Definition 13.** For a pair (x, σ) define a function $f_{x,\sigma} : \mathbb{F}^m \to \mathbb{F}$ as follows:

$$f_{x,\sigma}(y) = \mathop{\texttt{argmax}}_{C \sim \mathcal{C}_y \cap \mathcal{C}_{x,\sigma}} \{T(C)(y)\}.$$

If $\mathcal{C}_y \cap \mathcal{C}_{x,\sigma} = \emptyset$, define $f_{x,\sigma}(y)$ arbitrarily.

▶ **Claim 14.** *For an $(\frac{\epsilon}{2}, \gamma)$ excellent pair (x, σ),*

$$\mathop{\mathbf{Pr}}_{C \sim \mathcal{C}_{x,\sigma}, y \sim C}[f_{x,\sigma}(y) = T(C)(y)] \geq 1 - \gamma.$$

Proof. Fix an $(\frac{\epsilon}{2}, \gamma)$ excellent pair (x, σ), and denote $f = f_{x,\sigma}$. If we pick a uniform $C_1 \in \mathcal{C}_{x,\sigma}$, then $y \in C_1$ such that $y \neq x$, and a uniform $C_2 \in \mathcal{C}_{x,\sigma} \cap \mathcal{C}_y$, then

$$\mathop{\mathbf{Pr}}_{C_1,y,C_2}[T(C_1)(y) \neq T(C_2)(y)] \leq \mathop{\mathbf{Pr}}_{C_1,y,C_2}[T(C_1)_{|\ell(x,y)} \neq T(C_2)_{|\ell(x,y)}] \leq \gamma,$$

since (x, σ) is $(\frac{\epsilon}{2}, \gamma)$ excellent.

For each y, denote $\gamma_y = \mathbf{Pr}_{C_1,C_2 \sim \mathcal{C}_{x,\sigma} \cap \mathcal{C}_y}[T(C_1)(y) \neq T(C_2)(y)]$. From the above we get that $\mathbb{E}_y[\gamma_y] \leq \gamma$, where y is distributed according to it's weight in $\mathcal{C}_{x,\sigma}$. For each y,

$$\begin{aligned}
1 - \gamma_y &= \sum_{\theta \in \mathbb{F}} \mathop{\mathbf{Pr}}_{C \sim \mathcal{C}_{x,\sigma} \cap \mathcal{C}_y}[T(C)(y) = \theta]^2 \\
&\leq \mathop{\mathbf{Pr}}_{C \sim \mathcal{C}_{x,\sigma} \cap \mathcal{C}_y}[T(C)(y) = f(y)] \sum_{\theta \in \mathbb{F}} \mathop{\mathbf{Pr}}_{C \sim \mathcal{C}_{x,\sigma} \cap \mathcal{C}_y}[T(C)(y) = \theta] \\
& && (f(y) \text{ is the most frequent value}) \\
&\leq \mathop{\mathbf{Pr}}_{C \sim \mathcal{C}_{x,\sigma} \cap \mathcal{C}_y}[T(C)(y) = f(y)].
\end{aligned}$$

Since it is true for each y, it is also true when taking expectation over y, for any distribution:

$$\mathop{\mathbf{Pr}}_{C\sim\mathcal{C}_{x,\sigma},y\sim C}[f(y)=T(C)(y)]=\mathop{\mathbf{E}}_{y}\left[\mathop{\mathbf{E}}_{C\sim\mathcal{C}_{x,\sigma}\cap\mathcal{C}_y}[\mathbb{I}(T(C)(y)=f(y))]\right]\geq\mathop{\mathbf{E}}_{y}[1-\gamma_y]\geq 1-\gamma.$$

In expectation, each y is chosen with probability proportional to it's weight in $\mathcal{C}_{x,\sigma}$, as before. ◀

Proof of Lemma 11. From Claim 12 we know that the probability of (x,σ) to be $(\frac{\epsilon}{2},\gamma)$-excellent is at least $\frac{\epsilon}{3}$. Since x is chosen uniformly, it means that for at least $\frac{\epsilon}{3}$ of the inputs $x\in\mathbb{F}^m$ there exists some $\sigma\in\mathbb{F}$ such that (x,σ) is excellent. If there is more than one such σ choose one arbitrarily.

Fixing an excellent x, let σ be the value such that (x,σ) is excellent. For this σ, $\mathbf{Pr}_{C\in\mathcal{C}_x}[C\in\mathcal{C}_{x,\sigma}]\geq\frac{\epsilon}{2}$. From Claim 14, $\mathbf{Pr}_{C\sim\mathcal{C}_{x,\sigma},y\sim C}[f_{x,\sigma}(y)=T(C)(y)]\geq 1-\gamma$. By averaging, at least half of the cubes $C\in\mathcal{C}_{x,\sigma}$ satisfy $\mathbf{Pr}_{y\sim C}[f_{x,\sigma}(y)=T(C)(y)]\geq 1-2\gamma$. For all these cubes $T(C)\overset{2\gamma}{\approx}f_{x,\sigma}$, and they are at least $\frac{\epsilon}{4}$ fraction of the cubes in $\mathcal{C}_x$. ◀

3.2 Global Structure

In this section, we prove the following lemma:

▶ **Lemma 15** (Global Structure). *Let T be a cubes table that passes Test 1 with probability at least $\epsilon=\Omega(\frac{d^4}{\sqrt{q}})$, then for every $\gamma=\Omega(\frac{1}{d^3})$, there exists an $(\frac{\epsilon}{2},\gamma)$-excellent x such that $f=f_x:\mathbb{F}^m\to\mathbb{F}$ satisfies*

$$\mathop{\mathbf{Pr}}_{C}[T(C)\overset{32\gamma}{\approx}f_{|C}]\geq\frac{\epsilon}{16}.$$

Let $X^\star\subseteq\mathbb{F}^m$ the set of $(\frac{\epsilon}{2},\gamma)$ excellent points.

The main idea in the proof of the global structure, is showing that there exist many pairs of excellent points $x,y\in X^\star$, such that for many cubes C, the $T(C)$ is similar both to f_x and to f_y (Claim 17). If this is the case, then the functions f_x,f_y must be very similar (Claim 18). Finally, the lemma is proven by averaging and finding a single x such that f_x agrees simultaneously with many of the f_y's and their supporting cubes.

▶ **Definition 16** (Supporting cubes). For any excellent $x\in X^\star$, we denote by F_x the set of cubes "supporting" f_x,

$$F_x=\left\{C\in\mathcal{C}_x\,\middle|\,T(C)\overset{2\gamma}{\approx}f_{x|_C}\right\}.$$

▶ **Claim 17.** *Let $\mathcal{D}$ be the following process: choose $x,y\in X^\star$ independently and uniformly at random, let C be a random cube containing both x and y. Then*

$$\mathop{\mathbf{Pr}}_{x,y,C\sim D}[C\in F_x\cap F_y]\geq\frac{\epsilon^2}{26}.$$

Proof. Since each $x\in X^\star$ is excellent, we know from the local structure lemma, Lemma 11, that $\mathbf{Pr}_{C\sim\mathcal{C}_x}[C\in F_x]\geq\frac{\epsilon}{4}$. This is of course also true when taking a uniform $x\in X^\star$, thus, $\mathbf{Pr}_{x\sim X^\star,C\sim\mathcal{C}_x}[C\in F_x]\geq\frac{\epsilon}{4}$.

From Lemma 9(4) , the inclusion graph $G=G(\mathbb{F}^m,\mathcal{C})$ has $\lambda(G)=\lambda\leq(1+o(1))\frac{1}{q^{3/2}}$. Denote the measure of $X^\star$ by μ, from Lemma 11, $\mu\geq\frac{\epsilon}{3}$. Hence, by the application of Lemma 6 on the graph G with $A=\mathcal{C},B=\mathbb{F}^m$ and $B'=X^\star$, we get

$$\left|\mathop{\mathbf{Pr}}_{x\sim X^\star,C\sim\mathcal{C}_x}[C\in F_x]-\mathop{\mathbf{Pr}}_{C\sim\mathcal{C},x\sim C\cap X^\star}[C\in F_x]\right|\leq\frac{\lambda}{\sqrt{\mu}}\leq\frac{2\lambda}{\sqrt{\epsilon}}.\tag{2}$$

For each $C \in \mathcal{C}$, let $p_C = \mathbf{Pr}_{x \sim C \cap X^\star}[C \in F_x]$, this measures for every cube C how many points $x \in C$ are such that $f_{x|_C} \overset{2\gamma}{\approx} T(C)$. In this notation, (2) implies $\mathbf{E}_C[p_C] \geq \frac{\epsilon}{4} - \frac{2\lambda}{\sqrt{\epsilon}} \geq \frac{\epsilon}{5}$. We can use this to bound the probability of the event $C \in F_x \cap F_y$ by first choosing C, then two independent points in $C \cap X^\star$,

$$\mathop{\mathbf{Pr}}_{\substack{C \sim \mathcal{C} \\ x,y \sim C \cap X^\star}}[C \in F_x \cap F_y] = \mathop{\mathbf{E}}_{C}[p_C^2] \geq \left(\mathop{\mathbf{E}}_{C}[p_C]\right)^2 \geq \frac{\epsilon^2}{25}.$$

We observe that this distribution is very similar to the required distribution D. The only difference is that here we first pick $C \in \mathcal{C}$ and then two excellent points in C, whereas in D we first pick two points in $X^\star$ and then a common neighbor C. The graph G satisfies that every two distinct points $x, y \in \mathbb{F}^m$ have exactly the same number of common neighbors. Therefore, we can use Lemma 8 on the graph G with $A = \mathcal{C}, B = \mathbb{F}^m$ and $B' = X^\star$ to get

$$\left| \mathop{\mathbf{Pr}}_{\substack{C \sim \mathcal{C} \\ x,y \sim C \cap X^\star}}[C \in F_x \cap F_y] - \mathop{\mathbf{Pr}}_{x,y,C \sim D}[C \in F_x \cap F_y] \right| \leq \frac{2\lambda}{\mu} + \frac{1}{\mu^2 d_A} + \frac{1}{\mu^2 |B|} \leq \frac{6\lambda}{\epsilon} + \frac{9}{q^m \epsilon^2} + \frac{9}{q^3 \epsilon^2}.$$

Recall that $\lambda \leq (1 + o(1))\frac{1}{q^{3/2}}$ and since $\epsilon = \Omega(\frac{d^4}{\sqrt{q}})$, we conclude that $\mathbf{Pr}_{x,y,C \sim D}[C \in F_x \cap F_y] \geq \frac{\epsilon^2}{25} - \frac{6\lambda}{\epsilon} - \frac{9}{q^m \epsilon^2} - \frac{9}{q^3 \epsilon^2} \geq \frac{\epsilon^2}{26}$. ◀

▶ **Claim 18.** *Let $x \neq y \in X^\star$, and let ℓ be the line containing x and y, if $\mathbf{Pr}_{C \sim \mathcal{C}_\ell}[C \in F_x \cap F_y] \geq \frac{\epsilon^2}{100}$ then $f_x \overset{5\gamma}{\approx} f_y$.*

Proof. Consider the graph $G = G(\mathbb{F}^m \setminus \ell, \mathcal{C}_\ell)$. This is a bi-regular bipartite graph, and by Lemma 9(3) it has $\lambda = \lambda(G) \leq (1 + o(1))\frac{1}{\sqrt{q}}$. Let $F = F_x \cap F_y$. By assumption, F has measure at least $\frac{\epsilon^2}{100}$ inside $\mathcal{C}_\ell$.

We denote by $E' \subset E$ the edges of G that indicate agreement with both f_x and f_y,

$$E' = \{(z, C) \mid T(C)(z) = f_x(z) = f_y(z)\}.$$

Every cube $C \in F$ has $1 - 2\gamma$ of the points $z \in C$ satisfying $T(C)(z) = f_x(z)$ and $1 - 2\gamma$ of the points satisfying $T(C)(z) = f_y(z)$. By a union bound we get $\mathbf{Pr}_{C \in F, z \in N(C)}[(z, C) \in E'] \geq 1 - 4\gamma$. By Lemma 6 on G when $A = \mathbb{F}^m \setminus \ell, B = \mathcal{C}_\ell, B' = F$,

$$\left| \mathop{\mathbf{Pr}}_{C \sim F, z \sim N(C)}[(z, C) \in E'] - \mathop{\mathbf{Pr}}_{z, C \sim N(z) \cap F}[(z, C) \in E'] \right| \leq \frac{20\lambda}{\epsilon},$$

which means that $\mathbf{Pr}_{z \sim \mathbb{F}^m, C \sim N(z) \cap F}[(z, C) \in E'] \geq 1 - 4\gamma - \frac{20\lambda}{\epsilon} \geq 1 - 5\gamma$. By the definition of E', for each point $z \in \mathbb{F}^m$ that has an adjacent edge in E', $f_x(z) = f_y(z)$. This means that

$$\mathop{\mathbf{Pr}}_{z}[f_x(z) = f_y(z)] \geq \mathop{\mathbf{Pr}}_{z}[\exists C \text{ s.t. } (z, C) \in E'] \geq \mathop{\mathbf{Pr}}_{z, C \sim N(z) \cap F}[(z, C) \in E'] \geq 1 - 5\gamma.$$ ◀

The above claim showed that if two functions have a large set of cubes on which they almost agree then these functions are similar. In order to prove the global structure, we also need to show that in this case, most of $C \in F_y$ will also be close to f_x.

▶ **Claim 19.** *Let $x, y \in X^\star$ such that $f_x \overset{5\gamma}{\approx} f_y$, then*

$$\mathop{\mathbf{Pr}}_{C \sim F_y}[T(C) \overset{32\gamma}{\approx} f_{x|_C}] \geq \frac{1}{2}.$$

Note that the function f_x may not be a low degree polynomial, so $T(C) \overset{32\gamma}{\approx} f_{x|_C}$ doesn't imply equality.

Proof. Let $G = G(\mathbb{F}^m \setminus \{y\}, \mathcal{C}_y)$, by Claim 9(5) it has $\lambda = \lambda(G) \approx \frac{1}{q}$. First, we denote by E'_y the following set of edges,

$$E'_y = \{(z, C) \mid T(C)(z) = f_y(z)\}.$$

For each $C \in F_y$, we know that $\mathbf{Pr}_{z \in N(C)}[(z, C) \in E'_y] \geq 1 - 2\gamma$. From Lemma 6 on G when $A = \mathbb{F}^m \setminus y, B = \mathcal{C}_y, B' = F_y$, we know that

$$\left| \mathop{\mathbf{Pr}}_{C \sim F_y, z \sim N(C)}[(z, C) \in E'_y] - \mathop{\mathbf{Pr}}_{z, C \in N(z) \cap F_y}[(z, C) \in E'_y] \right| \leq \frac{4\lambda}{\epsilon},$$

since the measure of F_y is at least $\frac{\epsilon}{4}$. This implies that $\mathbf{Pr}_{z, C \in N(z) \cap F_y}[(z, C) \in E'_y] \geq 1 - 3\gamma$.

We define a second set of edges, E'_x to be the same only for f_x,

$$E'_x = \{(z, C) \mid T(C)(z) = f_x(z)\}.$$

We notice that if z is a point such that $f_x(z) = f_y(z)$, then $(z, C) \in E'_y \Rightarrow (z, C) \in E'_x$.

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{z, C \sim N(z) \cap F_y}[(z, C) \in E'_x] \geq & \mathop{\mathbf{Pr}}_{z}[f_x(z) = f_y(z)] \cdot \mathop{\mathbf{Pr}}_{z, C \sim N(z) \cap F_y}[(z, C) \in E'_y \mid f_x(z) = f_y(z)] \\
\geq & (1 - 5\gamma) \cdot \mathop{\mathbf{Pr}}_{z, C \sim N(z) \cap F_y}[(z, C) \in E'_y \mid f_x(z) = f_y(z)] \\
& \qquad \text{(since } f_x \overset{5\gamma}{\approx} f_y\text{)} \\
\geq & (1 - 5\gamma) \cdot \left(\mathop{\mathbf{Pr}}_{z, C \sim N(z) \cap F_y}[(z, C) \in E'_y] - 5\gamma \right) \\
\geq & 1 - 15\gamma.
\end{aligned}$$

Therefore, we can use Lemma 6 again on the same graph G and set F_y, now with the edge set E'_x, to conclude that

$$\mathop{\mathbf{Pr}}_{C \sim F_y, z \sim N(C)}[(z, C) \in E'_x] \geq \mathop{\mathbf{Pr}}_{z, C \sim N(z) \cap F_y}[(z, C) \in E'_x] - \frac{4\lambda}{\epsilon} \geq 1 - 16\gamma,$$

By averaging, at least half of $C \in F_y$ satisfies $T(C) \overset{32\gamma}{\approx} f_{x|_C}$. ◀

We are now ready to prove the global structure.

Proof of Lemma 15. Let T be the cubes table that passes Test 1 with probability at least $\epsilon = \Omega(\frac{d^4}{\sqrt{q}})$. From the local structure, Lemma 11, we know that there exists a set $X^\star$ of excellent points, such that each $x \in X^\star$ has a function f_x, and $|F_x| \geq \frac{\epsilon}{4} |\mathcal{C}_x|$.

From Claim 17, we know that $\mathbf{Pr}_{x,y,C \sim D}[C \in F_x \cap F_y] \geq \frac{\epsilon^2}{26}$, when x, y are chosen uniformly from $X^\star$ and C is a common neighbor. Therefore, there must be $x \in X^\star$ such that $\mathbf{Pr}_{y \sim X^\star, C \sim N(x) \cap N(y)}[C \in F_x \cap F_y] \geq \frac{\epsilon^2}{26}$.

Fix such $x \in X^\star$, and let X' be the set of $y \in X^\star$ such that $|F_x \cap F_y| \geq \frac{\epsilon^2}{100} |\mathcal{C}_\ell|$. By averaging, $|X'| \geq \frac{\epsilon^2}{100} |X^\star| \geq \frac{\epsilon^3}{400} |\mathbb{F}|^m$.

By Claim 18, for all $y \in X'$, $f_y \overset{5\gamma}{\approx} f_x$. For each $y \in X'$, let

$$F'_y = \{C \in F_y \mid T(C) \overset{32\gamma}{\approx} f_{x|_C}\}.$$

At this point we have a large collection of y's and for each one a large collection of cubes F'_y such that all of these support the same function f_x. It is immediate that f_x is supported by some poly(ϵ) fraction of all of the cubes. Since we are aiming for a better quantitative bound of $\Omega(\epsilon)$ fraction of $\mathcal{C}$, we will rely on the expansion once more.

In order to finish the proof, we need to show that $\left|\cup_{y\in X'} F'_y\right| \geq \frac{\epsilon}{16}|\mathcal{C}|$.

Let $G = G(\mathbb{F}^m, \mathcal{C})$, by Lemma 9(4) $\lambda(G) \leq q^{-\frac{3}{2}}$. We use X' as the set of vertices, and define

$$E' = \{(y, C) \mid T(C) \overset{32\gamma}{\approx} f_{x|_C}\}.$$

By Lemma 6 on G with $A = \mathcal{C}, B = \mathbb{F}^m, B' = X'$,

$$\left| \Pr_{y\sim X', C\sim N(y)}[(y, C) \in E'] - \Pr_{C\sim\mathcal{C}, y\sim N(C)\cap X'}[(y, C) \in E'] \right| \leq \frac{20\lambda}{\sqrt{\epsilon^3}} \leq \frac{20q^{-\frac{3}{2}}}{q^{-\frac{3}{4}}} \leq 20q^{-\frac{3}{4}} \leq \frac{\epsilon}{16},$$

where we used the fact that $\epsilon \geq \frac{1}{\sqrt{q}}$.

Claim 19 lets us bound the first term on the left, since for each $y \in X'$, $\mathbf{Pr}_{C\sim N(y)}[C \in F'_y] \geq \frac{1}{2}\mathbf{Pr}_{C\sim N(y)}[C \in F_y] \geq \frac{\epsilon}{8}$. Thus,

$$\Pr_{C\sim\mathcal{C}, y\sim N(C)\cap X'}[(y, C) \in E'] \geq \frac{\epsilon}{8} - \frac{\epsilon}{16} = \frac{\epsilon}{16}.$$

We notice that a cube with even a single adjacent edge in E' satisfies $T(C) \overset{32\gamma}{\approx} f_{x|_C}$, so we are done. ◀

3.3 Low Degree

The last step is to prove that the global function discovered in the previous section can be modified to make it a low degree function, while still maintaining large support for it among the cubes.

▶ **Theorem 20** (Theorem 2 restated). *For every d and large enough prime power q and every $m \geq 3$ the following holds. Let T be a cubes table that passes Test 1 with probability at least $\epsilon = \Omega(\frac{d^4}{\sqrt{q}})$, then there exist a degree d polynomial $g : \mathbb{F}^m \to \mathbb{F}$ such that $T(C) = g|_C$ on an $\Omega(\epsilon)$ fraction of the cubes.*

From Lemma 15, we get a function f such that $\Omega(\epsilon)$ of the cubes have $T(C) \approx f_{|_C}$. In this section, we will show that this function f is close to a degree d polynomial g. Afterwards, we also need to show that $\Omega(\epsilon)$ of the cubes satisfies $T(C) = g_{|_C}$

To show the first part, we will use a robust characterization of low degree polynomials given by Rubinfeld and Sudan.

▶ **Theorem 21** ([12, Theorem 4.1]). *Let $f : \mathbb{F}^m \to \mathbb{F}$ be a function, and let $N_{y,h} = \{y+i(h-y) \mid i \in \{0, \ldots, d+1\}\}$, if f satisfies*

$$\Pr_{y,h\in\mathbb{F}^m}[\exists \deg d \text{ polynomial } p \textbf{ s.t. } p_{|_{N_{y,h}}} = f_{|_{N_{y,h}}}] \geq 1 - \delta,$$

for $\delta \leq \frac{1}{2(d+2)^2}$, then there exists a degree d polynomial g such that $f \overset{2\delta}{\approx} g$.

For completeness, we present proof of the above theorem in Appendix C.

▶ **Claim 22.** *Fix any* $\gamma \leq \frac{1}{100(d+2)^3}$*, let* $f : \mathbb{F}^m \to \mathbb{F}$ *and* $x \in \mathbb{F}^m$ *such that*

$$\Pr_{C \in \mathcal{C}_x}[T(C) \overset{32\gamma}{\approx} f_{|C}] \geq \frac{\epsilon}{4},$$

then exists a degree d *polynomial* g *such that* $f \overset{84d\gamma}{\approx} g$*.*

Proof. Denote by $F \subseteq \mathcal{C}_x$ the following set

$$F = \{C \in \mathcal{C}_x \mid T(C) \overset{32\gamma}{\approx} f_{|C}\}.$$

Our first goal is to show that for nearly all lines, f agrees with a low degree function on almost all of the points of the line.

Fix $C \in F$, if we pick a uniform $\ell \subset C$ we expect that $T(C)_\ell \overset{O(\gamma)}{\approx} f_{|\ell}$. Using the spectral properties we show that almost all lines satisfy this property. Let $G_C = G(A \cup B, E)$ be the following bipartite inclusion graph where A is all the points in C, and B is all the affine lines in C. Let $A' \subset A$ be $A' = \{y \in A \mid T(C)(y) \neq f(y)\}$, and $B' \subset B$ be $B' = \{\ell \in B \mid |N(\ell) \cap A'| \geq 40\gamma\,|N(\ell)|\}$. From Lemma 9(6) with $m = 3$ (we apply the lemma where "$\mathbb{F}^m$" is the cube C), $\lambda_C = \lambda(G_C) \leq \frac{2}{\sqrt{q}}$. We apply Lemma 6 on G_C and the set B', where the set of edges is all the edges adjacent to A':

$$\left| \Pr_{y \in A, \ell \in N(y) \cap B'}[y \in A'] - \Pr_{\ell \in B', y \in N(\ell)}[y \in A'] \right| \leq \frac{\lambda_C}{\sqrt{\frac{|B'|}{|B|}}}.$$

We notice that $\mathbf{Pr}_{y \in A}[y \in A'] \leq 32\gamma$. By the definition of B', $\mathbf{Pr}_{\ell \in B', y \in N(\ell)}[y \in A'] \geq 40\gamma$. Therefore $|B'| \leq \left(\frac{\lambda_C}{8\gamma}\right)^2 |B| < \gamma\,|B|$.

We have shown that for every cube $C \in F$, almost all lines in it satisfy $T(C)_\ell \overset{40\gamma}{\approx} f_{|\ell}$. Now we need to show that the set F is large enough to cover $(1 - O(\gamma))$ of all the lines in $\mathcal{L}$. The inclusion graph $G = G(\mathcal{L} \setminus \mathcal{L}_x, \mathcal{C}_x)$ has $\lambda = \lambda(G) \leq \frac{1}{\sqrt{q}}$, by Lemma 9(1). We denote by E' the set of edges (ℓ, C) such that $T(C)_{|\ell} \overset{40\gamma}{\approx} f_{|\ell}$. As we've seen above, for every $C \in F$, $\mathbf{Pr}_{\ell \in N(C)}[(\ell, C) \in E'] \geq 1 - \gamma$.

By Lemma 6 on G, with $A = \mathcal{L} \setminus \mathcal{L}_x, B = \mathcal{C}_x, B' = F$,

$$\left| \Pr_{\ell, C \sim N(\ell) \cap F}[(\ell, C) \in E'] - \Pr_{C \sim F, \ell \sim C}[(\ell, C) \in E'] \right| \leq \frac{\lambda}{\sqrt{\epsilon}} \leq \gamma,$$

which means that

$$\Pr_{\ell}[\exists C \text{ s.t. } (\ell, C) \in E'] \geq \Pr_{\ell, C \sim N(\ell) \cap F}[(\ell, C) \in E'] \geq 1 - 2\gamma.$$

This means that for $1 - 2\gamma$ of the lines in $\mathcal{L}$, f agrees with a degree d function on $1 - 40\gamma$ fraction of the points of each line.

We are very close to being able to apply the low degree test of Rubinfeld and Sudan [12], that works in the high soundness regime. For this, we need to move to neighborhoods. For $y, h \in \mathbb{F}^m$, we define the neighborhood of y, h,

$$N_{y,h} = \{y + i(h - y) \mid 0 \leq i \leq d + 1\}.$$

Notice that $N_{y,h} \subset \ell(y, h)$. We show that on almost all of the neighborhoods $N_{y,h}$, the function $f_{|N_{y,h}}$ equals a degree d polynomial, by showing that for almost all $N_{y,h}$, there exists some cube C such that $f_{|N_{y,h}} = T(C)_{|N_{y,h}}$ ($T(C)$ is a degree d polynomial).

Picking a random neighborhood $N_{y,h}$ is equivalent to picking a random line $\ell \in \mathcal{L}$ and then uniform $y, h \in \ell$. We have already showed that almost all lines $\ell \in \mathcal{L}$, there exists a cube C such that $T(C)_\ell \overset{\Omega(\gamma)}{\approx} f_{|\ell}$.

Now we can bound the same probability over neighborhoods

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{y,h\sim\mathbb{F}^m}[\exists C \text{ s.t. } f(N_{y,h}) = T(C)(N_{y,h})] \\
&\quad \geq \mathop{\mathbf{Pr}}_{\ell}[\exists C \text{ s.t. } (\ell, C) \in E'] \cdot \mathop{\mathbf{Pr}}_{\ell,y,h\sim\ell}[f(N_{y,h}) = T(C)(N_{y,h}) \mid \exists C \text{ s.t. } (\ell, C) \in E'] \\
&\quad \geq (1-2\gamma) \mathop{\mathbf{Pr}}_{\ell,y,h\sim\ell}[f(N_{y,h}) = T(C)(N_{y,h}) \mid \exists C \text{ s.t. } (\ell, C) \in E'] \\
&\quad \geq (1-2\gamma)(1-(d+2)\cdot 40\gamma), \qquad (3) \\
&\quad \geq 1 - 42d\gamma,
\end{aligned}$$

where (3) is due to union bound on the neighborhoods inside ℓ. Therefore, the function f equals a degree d polynomial on $(1-42d\gamma)$ of the neighborhoods. Since $\gamma \leq 100(d+2)^{-3}$, by Theorem 21, we get that there exists a degree d polynomial g, such that $f \overset{84d\gamma}{\approx} g$. ◀

Proof of Theorem 20. Fix the cubes table T, and let $f : \mathbb{F}^m \to \mathbb{F}$ be the function promised from Lemma 15. This function satisfies the conditions of Claim 22, so there exists a degree d polynomial g such that $f \overset{84d\gamma}{\approx} g$.

Since g is a degree d polynomial, for every cube C either $T(C) = g_{|C}$, or else they are very different. Let G be the inclusion graph $G = G(\mathbb{F}^m, \mathcal{C})$, and let

$$F = \{C \in \mathcal{C} \mid T(C) \overset{32\gamma}{\approx} f_{|C}\}$$

From Lemma 15, the measure of F is at least $\frac{\epsilon}{16}$, let A' be the set of points on which $f \neq g$. By Lemma 9(4), $\lambda(G) \leq q^{-\frac{3}{2}}$. We use Lemma 6 on G with $A = \mathbb{F}^m, B = \mathcal{C}, B' = F$,

$$\left| \mathop{\mathbf{Pr}}_{C\in F, y\in N(C)}[y \in A'] - \mathop{\mathbf{Pr}}_{y, C\in N(y)\cap F}[y \in A'] \right| \leq \frac{q^{-\frac{3}{2}}}{\epsilon} \leq \gamma$$

We know that $\mathbf{Pr}_{y,C\in N(y)\cap F}[y \in A'] \leq \mathbf{Pr}_y[y \in A'] \leq 84d\gamma$, which implies that $\mathbf{Pr}_{C\in F, y\in N(C)}[y \in A'] \leq 85d\gamma$.

By averaging, for at least half of the cubes $C \in F$, $\mathbf{Pr}_{y\in C}[y \in A'] \leq 200d\gamma \leq \frac{1}{2}$. For all these cubes $T(C) = g_{|C}$, because $\mathbf{Pr}_{y\in C}[T(C)(y) = g(y)] \geq \mathbf{Pr}_{y\in C}[T(C)(y) = f(y), y \notin A'] \geq 1 - 32\gamma - \frac{1}{2} > d/q$, and since $g_{|C}, T(C)$ are both degree d polynomials, they must be equal. ◀

▶ Remark. Instead of Theorem 21, we can use another similar characterization from [12], where the neighborhood is defined as $N_{y,h} = \{y + i(h-y) \mid i \in \{0, \ldots, 10d\}\}$. The advantage of using this new neighborhood is that we can conclude $f \overset{(1+o(1))\delta}{\approx} g$ as long as $\delta = O(1/d)$. This will help in reducing the exponent of d by 1 in our main theorem. We chose to use Theorem 21 for a self contained proof.

4 Comparing between different tests and their agreement parameter

There are many variants for the low degree test, in this section we look into equivalences between similar low degree agreement tests. We first prove the equivalence in a more general setting and as a corollary we get some interesting results.

Throughout this section, we will work over $\mathbb{F}^m$ where $\mathbb{F}$ is a field of size q and let $s \leq m/2$ be fixed. Also, let T denotes a table which maps every s dimensional affine subspace in $\mathbb{F}^m$ to a degree d polynomial. Let $\mathcal{A}^s$ denote the set of all s dimensional affine subspaces in $\mathbb{F}^m$. For $r < s$ and for $R \in \mathcal{A}^r$ let $\mathcal{A}^s_R \subseteq \mathcal{A}^s$ denote all subspaces in $\mathcal{A}^s$ which contain a particular subspace R,

$$\mathcal{A}^s_R = \{S \subset \mathbb{F}^m \mid \dim(S) = s, R \subseteq S\}.$$

For parameters $s > k \geq r$ consider the following test:

Test 3 Subspace agreement test : $\alpha_{sks(r)}$

1. Select $K \in \mathcal{A}^k$ u.a.r.
2. Pick $S_1, S_2 \in \mathcal{A}^s_K$ u.a.r.
3. Pick a r dimensional subspace $R \subseteq K$ u.a.r.
4. Accept iff $T(S_1)_{|R} = T(S_2)_{|R}$.

Let $\alpha_{sks(r)}(T)$ be the *agreement* of the table $T = (f_S)_{S \in \mathcal{A}^s}$, i.e. the probability of acceptance of the test.

When $r = k$ we simply denote the agreement as $\alpha_{sks}(T)$. With these notations, the success probability of Test 1 is denoted by $\alpha_{3,0,3}(T)$, and of Test 2 by $\alpha_{2,1,2}(T)$.

In this section, we prove the following main lemma.

▶ **Lemma 23.** *Let $0 \leq r < k < s \leq \frac{m}{2}$, we have*

$$\alpha_{srs}(T)\left(1 - \left(\frac{d}{q}\right)^{r+1}\right) \leq \alpha_{sks}(T) \leq \alpha_{srs}(T) + (1 + o(1))q^{-(s-2k+r+1)},$$

From Lemma 23, we can deduce the following corollary,

▶ **Corollary 24.** *Let $\alpha_{\mathcal{C}\ell\mathcal{C}}(T) = \alpha_{3,1,3}(T)$ be the success probability of Test 3 with $s = 3, k = r = 1$, i.e checking consistency of two cubes that intersect on a line. Then for every cubes table T,*

$$\alpha_{\mathcal{C}x\mathcal{C}}(T)\left(1 - \frac{d}{q}\right) \leq \alpha_{\mathcal{C}\ell\mathcal{C}}(T) \leq \alpha_{\mathcal{C}x\mathcal{C}}(T) + \frac{1}{q^2}(1 + o(1)).$$

The corollary implies that Theorem 2 holds if we modify the test as selecting two cubes u.a.r from a pair of cubes intersecting in a line and checking consistency on the whole line.

Using Lemma 23, we can also compare the Raz-Safra Plane vs. Plane agreement tests where planes intersect at a point and on a line. Recall that $\alpha_{\mathcal{P}\ell\mathcal{P}}(T)$ is the acceptance probability of Test 2. Invoking Lemma 23 with $s = 2$, $k = 1$ and $r = 0$, we get the following corollary.

▶ **Corollary 25** (Lemma 3 restated)**.** *Let T be a planes table, and let $\alpha_{\mathcal{P}x\mathcal{P}}(T)$ be the success probability of Test 3 with $s = 2, k = r = 0$, i.e two planes that intersects on a point. Let $\alpha_{\mathcal{P}\ell\mathcal{P}}(T)$ be the success probability of Test 2 from the introduction (two planes that intersects on a line), then*

$$\alpha_{\mathcal{P}x\mathcal{P}}(T)\left(1 - \frac{d}{q}\right) \leq \alpha_{\mathcal{P}\ell\mathcal{P}}(T) \leq \alpha_{\mathcal{P}x\mathcal{P}}(T) + \frac{1}{q}(1 + o(1)).$$

4.1 Proof of Lemma 23

We prove a few claims that together with the observation $\alpha_{sks(r)}(T) \geq \alpha_{sks}(T)$, prove the lemma.

The following claim shows that two distinct low degree polynomials agree on a random subspace of fixed dimension with very small probability.

▶ **Claim 26.** *Let $P_1, P_2 : \mathbb{F}^t \to \mathbb{F}$ be two distinct degree d polynomials. For $r \leq t$*

$$\Pr_{R \in \mathcal{A}^r}\left[(P_1)_{|R} \equiv (P_2)_{|R}\right] \leq \left(\frac{d}{q}\right)^{r+1}.$$

Proof. Consider the following way of choosing an r dimensional *affine* subspace from $\mathcal{A}^r$ uniformly at random: Pick $x_0, x_1, x_2, \ldots, x_r$ from $\mathbb{F}_q^t$ independently and u.a.r. Then pick a r dimensional affine subspace R containing $\{x_0 + \mathtt{span}(x_1, x_2, \ldots, x_r)\}$ u.a.r (R is determined by $x_0, x_1, x_2, \ldots, x_r$, unless $\dim \mathtt{span}(x_1, x_2, \ldots, x_r) < r$). It is easy to see that R is distributed uniformly in $\mathcal{A}^r$. Now, P_1 and P_2 agreeing on the whole subspace R implies that they agree on the points $\{x_0, x_0 + x_1, x_0 + x_2, \ldots, x_0 + x_r\}$ as all these points are contained in R. Therefore,

$$\begin{aligned}\Pr_{R \in \mathcal{A}^r}[(P_1)_{|R} \equiv (P_2)_{|R}] &\leq \Pr_{x_0, x_1, \ldots, x_r \sim \mathbb{F}^t}[P_1(x_0) = P_2(x_0) \wedge_{i=1}^r P_1(x_0 + x_i) = P_2(x_0 + x_i)] \\ &= \left(\Pr_{x \in \mathbb{F}_q^t}[P_1(x) = P_2(x)]\right)^{r+1} \leq \left(\frac{d}{q}\right)^{r+1},\end{aligned}$$

where the last inequality is because two different degree d polynomial agree on at most $\frac{d}{q}$ fraction of the points (Schwartz-Zippel lemma). ◀

▶ **Claim 27.** *Let $M_{m \times n}$ be the adjacency matrix of a bi regular bipartite graph G, and let f be a n-dimensional $\{0,1\}$ vector such that $\mathbf{E}[f] = \mu$. Then*

$$\langle Mf, Mf \rangle \leq \mu^2 + \lambda(G)^2 \mu.$$

Proof. Let $\mathbf{1}$ be the unit vector. We write f as $f = f_1 + f_1^\perp$ where f_1 is in the direction of $\mathbf{1}$, the singular vector with the maximal singular value, and $f_1^\perp$ is its orthogonal component. We note that $f_1 = \mu \mathbf{1}$, and hence $\langle f_1, f_1 \rangle = \mu^2$. Also,

$$\mu = \langle f, f \rangle = \langle f_1 + f_1^\perp, f_1 + f_1^\perp \rangle = \langle f_1, f_1 \rangle + \langle f_1^\perp, f_1^\perp \rangle \geq \langle f_1^\perp, f_1^\perp \rangle.$$

Using this we can bound:

$$\begin{aligned}\langle Mf, Mf \rangle =& \langle Mf_1 + Mf_1^\perp, Mf_1 + Mf_1^\perp) \rangle \\ =& \langle f_1, f_1 \rangle + \langle Mf_1^\perp, Mf_1^\perp \rangle \\ \leq& \mu^2 + \lambda(G)^2 \langle f_1^\perp, f_1^\perp \rangle \\ \leq& \mu^2 + \lambda(G)^2 \mu.\end{aligned}$$ ◀

▶ **Claim 28.** $\alpha_{sks(r)}(T) \geq \alpha_{srs}(T)$.

Proof. We start by fixing $R \in \mathcal{A}^r, \sigma \in \mathbb{F}^{q^r}$. For each k dimensional subspace $K \in \mathcal{A}_R^k$, denote by p_K the following probability $p_K = \mathbf{Pr}_{S \sim \mathcal{A}_K^s}[T(S)_{|R} \equiv \sigma]$. In this notation

$$\begin{aligned}\Pr_{\substack{K \sim \mathcal{A}_R^k \\ S_1, S_2 \sim \mathcal{A}_K^s}}[T(S_1)_{|R} \equiv T(S_2)_{|R} \equiv \sigma] &= \mathop{\mathbf{E}}_K[p_K^2] \\ &\geq \left(\mathop{\mathbf{E}}_K[p_K]\right)^2 = \Pr_{S_1, S_2 \sim \mathcal{A}_R^s}[T(S_1)_{|R} \equiv T(S_2)_{|R} \equiv \sigma]. \quad (4)\end{aligned}$$

Now, we average over R, σ to get $\alpha_{srs}(T)$ and $\alpha_{sks(r)}(T)$:

$$\alpha_{srs}(T) = \mathop{\mathbf{Pr}}_{\substack{R\sim\mathcal{A}^r \\ S_1,S_2\sim\mathcal{A}^s_R}}[T(S_1)_{|R} \equiv T(S_2)_{|R}]$$
$$= \mathop{\mathbf{E}}_{R\sim\mathcal{A}^r}\left[\sum_{\sigma\in\mathbb{F}^{q^r}} \mathop{\mathbf{Pr}}_{S_1,S_2\sim\mathcal{A}^s_R}[T(S_1)_{|R} \equiv T(S_2)_{|R} \equiv \sigma]\right]. \tag{5}$$

Picking a uniform $R \in \mathcal{A}^r$ then $K \in \mathcal{A}^k_R$ is the same as picking $K \in \mathcal{A}^k$ and then a random r dimensional subspace R in K, so by definition

$$\alpha_{sks(r)}(T) = \mathop{\mathbf{Pr}}_{\substack{R\sim\mathcal{A}^r, K\sim\mathcal{A}^k_R \\ S_1,S_2\sim\mathcal{A}^s_K}}[T(S_1)_{|R} \equiv T(S_2)_{|R}]$$
$$= \mathop{\mathbf{E}}_{R\sim\mathcal{A}^r}\left[\sum_{\sigma\in\mathbb{F}^{q^r}} \mathop{\mathbf{Pr}}_{\substack{K\sim\mathcal{A}^k_R \\ S_1,S_2\sim\mathcal{A}^s_K}}[T(S_1)_{|R} \equiv T(S_2)_{|R} \equiv \sigma]\right]. \tag{6}$$

Using (4), (5) and (6), we get $\alpha_{sks(r)}(T) \geq \alpha_{srs}(T)$. ◀

▶ **Claim 29.** $\alpha_{sks}(T) \geq \alpha_{sks(r)}(T)\left(1 - \left(\frac{d}{q}\right)^{r+1}\right)$.

Proof. By the definition of the agreement,

$$\alpha_{sks}(T) = 1 - \mathop{\mathbf{E}}_{K\sim\mathcal{A}^k}\left[\mathop{\mathbf{Pr}}_{S_1,S_2\sim\mathcal{A}^s_K}[T(S_1)_{|K} \neq T(S_2)_{|K}]\right],$$

and

$$\alpha_{sks(r)}(T) = 1 - \mathop{\mathbf{E}}_{K\sim\mathcal{A}^k}\left[\mathop{\mathbf{Pr}}_{\substack{R\sim K, \\ S_1,S_2\sim\mathcal{A}^s_K}}[T(S_1)_{|R} \neq T(S_2)_{|R}]\right],$$

where we use $R \sim K$ to denote a random r dimensional subspace in K. For every subspace $K \in \mathcal{A}^k$, $R \subseteq K$ is uniform and is independent of S_1, S_2.

$$\mathop{\mathbf{Pr}}_{\substack{R\sim K, \\ S_1,S_2\sim\mathcal{A}^s_K}}[T(S_1)_{|R} \neq T(S_2)_{|R}] = \mathop{\mathbf{Pr}}_{\substack{R\sim K, \\ S_1,S_2\sim\mathcal{A}^s_K}}[T(S_1)_{|K} \neq T(S_2)_{|K}, T(S_1)_{|R} \neq T(S_2)_{|R}]$$
$$= \mathop{\mathbf{Pr}}_{S_1,S_2\sim\mathcal{A}^s_K}[T(S_1)_{|K} \neq T(S_2)_{|K}] \cdot \mathop{\mathbf{Pr}}_{\substack{R\sim K, \\ S_1,S_2\sim\mathcal{A}^s_K}}[T(S_1)_{|R} \neq T(S_2)_{|R} \mid T(S_1)_{|K} \neq T(S_2)_{|K}]$$
$$\geq \mathop{\mathbf{Pr}}_{S_1,S_2\sim\mathcal{A}^s_K}[T(S_1)_{|K} \neq T(S_2)_{|K}] \cdot \left(1 - \left(\frac{d}{q}\right)^{r+1}\right).$$

The lower bound on the probability in the last inequality is as follows: the event $T(S_1)_{|K} \neq T(S_2)_{|K}$ implies that the degree d polynomials corresponding to $T(S_1)_{|K}$ and $T(S_2)_{|K}$ are distinct. Thus, using Claim 26 $\mathbf{Pr}_{R\sim K}[T(S_1)_{|R} \equiv T(S_2)_{|R}] \leq (d/q)^{r+1}$. Therefore, for a k dimensional subspace $K \in \mathcal{A}^k$,

$$\mathop{\mathbf{Pr}}_{\substack{R\sim K, \\ S_1,S_2\sim\mathcal{A}^s_K}}[T(S_1)_{|R} \neq T(S_2)_{|R}] \geq \mathop{\mathbf{Pr}}_{S_1,S_2\sim\mathcal{A}^s_K}[T(S_1)_{|K} \neq T(S_2)_{|K}]\left(1 - \left(\frac{d}{q}\right)^{r+1}\right).$$

Finally, taking the expectation of the inequality over K finishes the proof. ◀

We first state a lemma about an expansion of the kind of inclusion graphs which we will be dealing with in analyzing the Test 3, the proof of which appears in Appendix A.

▶ **Lemma 30.** *Let $r \leq k < s \leq \frac{m}{2}$ be integers, and let G be the inclusion graph $G = G(\mathcal{A}_R^k, \mathcal{A}_R^s)$ for a r dimensional subspace R, where $R \neq \emptyset$. Then,*

$$\lambda(G)^2 \leq (1 + o(1)) \cdot q^{-(s-2k+r+1)}.$$

▶ **Claim 31.** *$\alpha_{sks(r)}(T) \leq \alpha_{srs(r)}(T) + \lambda(G)^2$ where G is the inclusion graph $G = G(\mathcal{A}_R^k, \mathcal{A}_R^s)$ for an r dimensional subspace R.*

Proof. Fix an r dimensional affine subspace $R \in \mathcal{A}^r$. We prove the following inequality:

$$\mathop{\mathbf{Pr}}_{\substack{K \sim \mathcal{A}_R^k, \\ S_1, S_2 \sim \mathcal{A}_K^s}} [T(S_1)_{|R} \equiv T(S_2)_{|R}] \leq \mathop{\mathbf{Pr}}_{S_1, S_2 \sim \mathcal{A}_R^s} [T(S_1)_{|R} \equiv T(S_2)_{|R}] + \lambda(G)^2, \tag{7}$$

Note that this implies the claim if we take expectation over $R \in \mathcal{A}^r$. Towards proving (7), for each value $\sigma \in \mathbb{F}^{q^k}$, denote by $A_\sigma \subseteq \mathcal{A}_R^s$ the following set

$$A_\sigma = \{S \in \mathcal{A}_R^s \mid T(S)_{|R} \equiv \sigma\},$$

and $\mu_\sigma = \frac{|A_\sigma|}{|\mathcal{A}_R^s|}$. Let f_σ be the indicator function for A_σ, for $S \in A_\sigma$, $f_\sigma(S) = 1$. By definition

$$\mathop{\mathbf{Pr}}_{S_1, S_2 \sim \mathcal{A}_R^s} [T(S_1)_{|R} \equiv T(S_2)_{|R}] = \sum_\sigma \mu_\sigma^2. \tag{8}$$

Let $G = G(\mathcal{A}_R^k, \mathcal{A}_R^s)$ be the inclusion graph, and denote by $M \in \mathbb{R}^{|\mathcal{A}_R^k| \times |\mathcal{A}_R^s|}$ the normalized adjacency matrix, such that each entry is either 0 or $\frac{1}{\deg(K)}$ where $K \in \mathcal{A}_R^k$.

For each k dimensional subspace $K \in \mathcal{A}_R^k$, the value $(Mf_\sigma)_K$ is the fraction of K's neighbors in A_σ, $(Mf_\sigma)_K = \mathbf{Pr}_{S \sim \mathcal{A}_K^s}[S \in A_\sigma]$. Therefore, the inner product gives us the expected value:

$$\langle Mf_\sigma, Mf_\sigma \rangle = \mathop{\mathbf{E}}_{K \in \mathcal{A}_R^k} \left[\mathop{\mathbf{E}}_{S \in \mathcal{A}_K^s} [S \in A_\sigma]^2 \right] = \mathop{\mathbf{E}}_{K \in \mathcal{A}_R^k} \left[\mathop{\mathbf{E}}_{S_1, S_2 \in \mathcal{A}_K^s} [S_1, S_2 \in A_\sigma] \right].$$

Therefore

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{\substack{K \sim \mathcal{A}_R^k, \\ S_1, S_2 \sim \mathcal{A}_K^s}} [T(S_1)_{|R} \equiv T(S_2)_{|R}] &= \sum_\sigma \langle Mf_\sigma, Mf_\sigma \rangle \\
&\leq \sum_\sigma \mu_\sigma^2 + \lambda(G)^2 \mu_\sigma && \text{(using Claim 27)} \\
&= \mathop{\mathbf{Pr}}_{S_1, S_2 \sim \mathcal{A}_R^s} [T(S_1)_{|R} \equiv T(S_2)_{|R}] + \lambda(G)^2. && \text{(from (8))}
\end{aligned}$$

which proves (7). ◀

Claim 31 together with Lemma 30 gives us $\alpha_{sks}(T) \leq \alpha_{srs}(T) + (1 + o(1))q^{-2(s-2k+r+1)}$. Claim 28 and Claim 29 prove the other inequality, $\alpha_{srs}(T)\left(1 - \left(\frac{d}{q}\right)^{r+1}\right) \leq \alpha_{sks}(T)$.

References

1 Sanjeev Arora and Madhu Sudan. Improved low-degree testing and its applications. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 485–495. ACM, 1997.

2 Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, PCPs, and nonapproximability — towards tight results. *SIAM Journal on Computing*, 27(3):804–915, June 1998.

3 M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 73–83, 1990.

4 A.E. Brouwer, A.M. Cohen, and A. Neumaier. *Distance-regular graphs*. Ergebnisse der Mathematik und ihrer Grenzgebiete. Springer, 1989.

5 Irit Dinur and Elazar Goldenberg. Locally testing direct product in the low error range. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 613–622. IEEE, 2008.

6 Irit Dinur and Omer Reingold. Assignment testers: Towards combinatorial proofs of the PCP theorem. *SIAM Journal on Computing*, 36(4):975–1024, 2006. Special issue on Randomness and Computation.

7 Irit Dinur and David Steurer. Direct product testing. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 188–196. IEEE, 2014.

8 Oded Goldreich and Shmuel Safra. A combinatorial consistency lemma with application to proving the PCP theorem. In *RANDOM: International Workshop on Randomization and Approximation Techniques in Computer Science*. LNCS, 1997.

9 Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. New direct-product testers and 2-query PCPs. *SIAM Journal on Computing*, 41(6):1722–1768, 2012.

10 Dana Moshkovitz and Ran Raz. Sub-constant error low degree test of almost-linear size. *SIAM J. Computing*, 38(1):140–180, 2008.

11 Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability pcp characterization of np. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 475–484. ACM, 1997.

12 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

13 van der Waerden, E. Artin, E. Noether, and F. Blum. *Modern Algebra*, volume 1. Frederick Ungar Publishing, 1949.

A Spectral properties of Certain Inclusion Graphs

Let $G_{s,k}$ be the intersection graph where the vertex set is all *linear subspaces* of dimension s in $\mathbb{F}_q^m$ and $U \sim U'$ iff $\dim(U \cap U') = k$. We will use the $T_{s,k}$ to denote the *Markov operator* associated with a random walk on this graph. We will need following fact about eigenvalues of $T_{k,k-1}$.

▶ **Definition 32.** k-th q-ary Gaussian binomial coefficient $\begin{bmatrix} m \\ k \end{bmatrix}_q$ is given by

$$\begin{bmatrix} m \\ k \end{bmatrix}_q := \prod_{i=0}^{k-1} \frac{q^m - q^i}{q^k - q^i}.$$

As q is fixed throughout the article, we will omit the subscript from now on.

▶ **Fact 33.** *([4, Theorem 9.3.3]) Suppose* $1 \leq k \leq \frac{m}{2}$,

1. *The number of k dimensional linear subspaces in $\mathbb{F}_q^m$ is exactly $\begin{bmatrix} m \\ k \end{bmatrix}$.*
2. *The degree of $G_{k,k-1}$ is $q\begin{bmatrix} k \\ 1 \end{bmatrix}\begin{bmatrix} m-k \\ 1 \end{bmatrix}$.*
3. *The eigen values of $T_{k,k-1}$ are*

$$\lambda_j(T_{k,k-1}) = \frac{q^{j+1}\begin{bmatrix} k-j \\ 1 \end{bmatrix}\begin{bmatrix} m-k-j \\ 1 \end{bmatrix} - \begin{bmatrix} j \\ 1 \end{bmatrix}}{q\begin{bmatrix} k \\ 1 \end{bmatrix}\begin{bmatrix} m-k \\ 1 \end{bmatrix}},$$

with multiplicities $\begin{bmatrix} m \\ j \end{bmatrix} - \begin{bmatrix} m \\ j-1 \end{bmatrix}$ for $j = 0, 1, \ldots, k$. Asymptotically, $\lambda_j(T_{k,k-1}) = \Theta(q^{-j})$.

▶ **Claim 34.** *For any $1 \leq k \leq \frac{m}{2}$ and , we have $\left|\lambda_1(T_{k,k-2}) - \lambda_1(T_{k,k-1})^2\right| = (1 + o(1))\frac{1}{q^k}$.*

Proof. Consider a two-step random walk on the graph $G_{k,k-1}$. We will show that with very high probability, a two-step random walk on $G_{k,k-1}$ corresponds to a single step random walk on $G_{k,k-2}$. Let U_1, U_2, U_3 be the vertices from a two-step random walk on $G_{k,k-1}$. Note that conditioned on the event $\dim(U_1 \cap U_3) = k-2$, the distribution of (U_1, U_3) is exactly same as a single step random walk on $G_{k,k-2}$. We will upper bound the probability of the event $\dim(U_1 \cap U_3) \neq k-2$.

Let $w_1 = U_1 \cap U_2$ and $w_2 = U_2 \cap U_3$, we can describe the distribution of the two-step random walk as follows:

1. Choose a uniform k dimensional subspace U_2.
2. Choose two random $k-1$ dimensional subspaces, $w_1, w_2 \subset U_2$.
3. Choose a point $x_1 \in \mathbb{F}^m \setminus U_2$, and set $U_1 = \mathrm{span}(w_1, x_1)$.
4. Choose a point $x_2 \in \mathbb{F}^m \setminus U_2$, and set $U_3 = \mathrm{span}(w_2, x_2)$.

By definition, U_2 has $\begin{bmatrix} k \\ k-1 \end{bmatrix}$ subspaces of size $k-1$, therefore $\mathbf{Pr}_{w_1,w_2}[w_1 = w_2] = \frac{1}{\begin{bmatrix} k \\ k-1 \end{bmatrix}}$. In order to satisfy $\dim(U_1 \cap U_3) \neq k-2$ given that $w_1 \neq w_2$, the point x_2 should be in U_1. There are $q^k - q^{k-1}$ points in $U_1 \setminus U_2$, and therefore this probability equals $\frac{|U_1 \setminus U_2|}{|\mathbb{F}^m \setminus U_2|} = \frac{q^k - q^{k-1}}{q^m - q^k}$.

$$\begin{aligned}
\mathbf{Pr}[\dim(U_1 \cap U_3) \neq k-2] &= \mathbf{Pr}[w_1 = w_2] + \mathbf{Pr}[\dim(U_1 \cap U_3) \neq k-2 \wedge w_1 \neq w_2] \\
&= \frac{1}{\begin{bmatrix} k \\ k-1 \end{bmatrix}} + \left(1 - \frac{1}{\begin{bmatrix} k \\ k-1 \end{bmatrix}}\right)\mathbf{Pr}[\dim(U_1 \cap U_3) \neq k-2 \mid w_1 \neq w_2] \\
&= \frac{1}{\begin{bmatrix} k \\ k-1 \end{bmatrix}} + \left(1 - \frac{1}{\begin{bmatrix} k \\ k-1 \end{bmatrix}}\right) \cdot \frac{q^k - q^{k-1}}{q^m - q^k} =: \beta.
\end{aligned}$$

Thus, we have

$$T_{k,k-1}^2 = \beta\mathcal{N} + (1-\beta)T_{k,k-2},$$

where $\mathcal{N}$ is a Markov operator corresponding to the two-step random walk on $G_{k,k-1}$, conditioning on $\dim(U_1 \cap U_3) \neq k-2$. The claim follows as $\beta = (1 + o(1))1/q^k$. ◀

Following fact follows from the definition of $\lambda(G)$.

▶ **Fact 35.** *For a bi-regular bipartite graph $G(A, B)$, if T is a Markov operator associated with a random walk of length two starting from A (or B) then $\lambda(G)^2 = \lambda(T)$.*

We now prove Lemma 9.

▶ **Lemma 36** (Restatement of Lemma 9). *We have for every $m \geq 6$,*
1. *For $G_1(\mathcal{L} \setminus \mathcal{L}_x, \mathcal{C}_x)$, $\lambda(G_1) \approx \frac{1}{\sqrt{q}}$.*

2. *For* $G_2(\mathcal{L}_x, \mathcal{C}_x)$, $\lambda(G_2) \approx \frac{1}{q}$.
3. *For* $G_3(\mathbb{F}^m \setminus \ell, \mathcal{C}_\ell)$, $\lambda(G_3) \approx \frac{1}{\sqrt{q}}$.
4. *For* $G_4(\mathbb{F}^m, \mathcal{C})$, $\lambda(G_4) \approx \frac{1}{q^{3/2}}$.
5. *For* $G_5(\mathbb{F}^m \setminus \{x\}, \mathcal{C}_x)$, $\lambda(G_5) \approx \frac{1}{q}$.

And for every $m \geq 3$

6. *For* $G_6(\mathbb{F}^m, \mathcal{L})$, $\lambda(G_6) \approx \frac{1}{\sqrt{q}}$.

where $\approx$ *denotes equality up to a multiplicative factor of* $1 \pm o(1)$.

Proof. Suppose T is an $n \times n$ Markov operator which is a convex combination of a bunch of other Markov operators: $T = \sum_{i=1}^{k} \alpha_i T_i$ where $\alpha_i \geq 0$ and $\sum_{i=1}^{k} \alpha_i = 1$, and that both T and T_i's are regular. As the row sum of each Markov operator is 1, the largest eigenvalue is 1, since both T and T_i's are regular, the eigenvector of the largest eigenvalue is the all 1 vector. The second largest eigenvalue of T can be upper bounded by

$$\begin{aligned}\lambda(T) &:= \max_{\substack{v\in\mathbb{R}^n, \|v\|=1, \\ v\perp\mathbf{1}}} \|Tv\| \\ &= \max_{\substack{v\in\mathbb{R}^n, \|v\|=1, \\ v\perp\mathbf{1}}} \left\| \sum_{i=1}^{k} \alpha_i T_i \right\| \\ &\leq \sum_{i=1}^{k} \max_{\substack{v\in\mathbb{R}^n, \|v\|=1, \\ v\perp\mathbf{1}}} \|\alpha_i T_i\| = \sum_{i=1}^{k} \alpha_i \lambda(T_i).\end{aligned}$$

In proving the lemma, we repeatedly use the above simple fact to upper bound the eigenvalue.

1. Without loss of generality, we can assume $x = \mathbf{0}$. Let d_L and d_R denote the left and right degree of G_1 respectively. Fix a line ℓ, d_L is the number of cubes containing ℓ and not passing through $\mathbf{0}$. Every point $x \notin \text{span}(\ell, \mathbf{0})$ defines a cube $C = \text{span}(x, \mathbf{0}, \ell)$. Thus, the number of linear cubes containing ℓ equals $d_L = \frac{q^m - q^2}{q^3 - q^2}$, where the denominator is the overcounting factor, the number of points that give the same cube.
   Fix a linear cube C. The right degree is the number of lines in C not passing through the origin which is $\frac{\binom{q^3}{2}}{\binom{q}{2}} - \frac{q^3-1}{q-1}$, where the first term counts all possible lines in C (each two different points define a line, we divide by the double counting) and the second term counts all the lines in C that pass through the origin.
   Let T_1 be the Markov operator associated with a two-step random walk in G_1 starting from $\mathcal{C}_x$. Using Fact 35, in order to bound $\lambda(G_1)$ it is enough to bound the second largest eigenvalue of T_1. Since G_1 is bi-regular, the first eigenvector of T_1 is the all ones vector. For every cube C, the number of two-step walks starting from C is $d_L \cdot d_R$.
   If $\dim\{C_1 \cap C_2\} = 1$, then the two cubes intersection is only on a line. Since both cubes are linear, it means that this line goes through the origin, therefore it doesn't correspond to a vertex on the left side, and there is no walk $C_1 \to \ell \to C_2$, so $(T_1)_{C_1,C_2} = 0$. Of course, the same holds if $\dim\{C_1 \cap C_2\} = 0$.
   If $\dim\{C_1 \cap C_2\} = 2$, there there is a plane going through the origin in both C_1, C_2. The number of walks $C_1 \to \ell \to C_2$ equals the number of lines in this plane that don't contain the origin, $\mathbf{0}$. Each pair of distinct points on the plane correspond to a line, and we divide by the double counting. Therefore the number of lines in a plane equals $\frac{\binom{q^2}{2}}{\binom{q}{2}}$. We subtract from it the number of lines in a plane that contains $\mathbf{0}$, resulting in $\frac{\binom{q^2}{2}}{\binom{q}{2}} - \frac{q^2-1}{q-1} =: \beta$.

If $C_1 = C_2$, then exists a path $C_1 \to \ell \to C_2$ for every line ℓ adjacent to C_1, and there are d_R such lines.
Since T_1 is a Markov operator, we need to normalize the number of paths between C_1, C_2 by dividing in the total number of outgoing paths from C_1, which equals $d_R \cdot d_L$. Therefore,

$$(T_1)_{C_i,C_j} = \begin{cases} \frac{d_R}{d_R \cdot d_L}, & \text{if } C_i = C_j \\ \frac{\beta}{d_R \cdot d_L}, & \text{if } \dim\{C_1 \cap C_2\} = 2 \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

Thus, we can write T_1 as:

$$T_1 = \frac{1}{d_L} I + \frac{\beta}{d_R d_L} \cdot G_{3,2} = \frac{1}{d_L} I + \frac{\beta d'}{d_R d_L} \cdot T_{3,2},$$

where d' is the degree of a vertex in $G_{3,2}$. One can verify that T_1 is indeed a convex combination of two Markov operators I and $T_{3,2}$. Since $G_{3,2}$ is a regular graph, the second eigenvector of $T_{3,2}$ is also orthogonal to $\mathbf{1}$. Hence,

$$\begin{aligned} \lambda(G_1)^2 = \lambda(T_1) &= \max_{\substack{v \in \mathbb{R}^{|\mathcal{C}_x|}, v \perp \mathbf{1} \\ \|v\|=1}} \|T_1 v\| = \max_{\substack{v \in \mathbb{R}^{|\mathcal{C}_x|}, v \perp \mathbf{1} \\ \|v\|=1}} \left\| \left(\frac{1}{d_L} I + \frac{\beta d'}{d_R d_L} \cdot T_{3,2} \right) v \right\| \\ &= \frac{1}{d_L} + \frac{\beta d'}{d_R d_L} \cdot \lambda_1(T_{3,2}). \end{aligned} \tag{10}$$

We now just need to plug in the values of β, d' and $\lambda_1(T_{3,2})$. Using Fact 33, $\lambda_1(T_{3,2})$ is given by the following expression,

$$\lambda_1(T_{3,2}) = \frac{q^2 \begin{bmatrix} 2 \\ 1 \end{bmatrix} \begin{bmatrix} m-4 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix}}{q \begin{bmatrix} 2 \\ 1 \end{bmatrix} \begin{bmatrix} m-3 \\ 1 \end{bmatrix}} = (1 + o(1)) \frac{1}{q}.$$

As we have seen before, $d_R = \frac{\binom{q^3}{2}}{\binom{q}{2}} - \frac{q^3-1}{q-1} = (1 + o(1))q^4$, $d_L = \frac{q^m - q^2}{q^3 - q^2} = (1 + o(1))q^{m-3}$ and $\beta = \frac{\binom{q^2}{2}}{\binom{q}{2}} - \frac{q^2-1}{q-1} = (1 + o(1))q^2$. From Fact 33, $d' = (1 + o(1))q^{m-1}$. Thus,

$$\frac{1}{d_L} = (1 + o(1)) \frac{1}{q^{m-3}}, \qquad \frac{\beta d'}{d_R d_L} \lambda_1(T_{3,2}) = (1 + o(1)) \frac{1}{q}$$

Plugging these values in (10) gives $\lambda(G_1) = (1 + o(1)) \frac{1}{\sqrt{q}}$ as required.

2. This bound is implied from a more general Lemma 30 we prove below with $s = 3, k = 1$ and $r = 0$.
3. In this case, it will be easier to bound the eigenvalue of the Markov operator associated with a random walk of length two starting from $\mathbb{F}^m \setminus \ell$. Let T_3 be the Markov operator. Now, the path of length two starting from x looks like $x \to C \to y$. Thus, the cube C contains all points from the affine plane spanned by x and ℓ. Let $p(x, \ell)$ be the affine plane spanned by x and ℓ. We have $\mathbf{Pr}[y \in p(x, \ell)] = \frac{q^2 - q}{q^3 - q} \approx \frac{1}{q}$. If $y \notin p(x, \ell)$ then the distribution of y is uniform in $\mathbb{F}^m \setminus p(x, \ell)$. Thus, we have

$$T_3 = (1 - o(1)) \left(1 - \frac{1}{q}\right) J + (1 + o(1)) \frac{1}{q} \mathcal{N},$$

where J is a Markov operator associated with a complete graph on $\mathbb{F}^m \setminus \ell$, with self loops and $\mathcal{N}$ is an appropriate Markov operator. Thus, we have bound $\lambda(T_3) = (1 + o(1)) \frac{1}{q}$. Since $\lambda(G_3)^2 = \lambda(T_3)$, the bound follows.

4. Proof of this is along the same lines as (3). The Markov operator here (starting a walk from the left side) can be written as

$$T_4 = (1 \pm o(1))\frac{1}{q^3}I + \left((1 \pm o(1))(1 - \frac{1}{q^3})\right) J,$$

where I is an identity matrix. Thus $\lambda(T_4) = (1 \pm o(1))\frac{1}{q^3} = \lambda(G_4)^2$.

5. The proof of this item is also similar to (3), we look on the path of length 2 starting from the left side, i.e $y \to C \to z$, and let T_5 be the Markov operator. Let $\ell(x,y)$ be the line spanned by x, y (where x is the fixed point, $G_5(\mathbb{F}^m \setminus \{x\}, \mathcal{C}_x)$), then $\mathbf{Pr}[z \in \ell(x,y)] = \frac{|\ell(x,y)\setminus\{x\}|}{|C\setminus\{x\}|} = \frac{q-1}{q^3-1} \approx \frac{1}{q^2}$, let $\mathcal{N}$ be the appropriate Markov operator of the event that x, y, z are colinear, then

$$T_5 = (1 - o(1))\left(1 - \frac{1}{q^2}\right) J + (1 + o(1))\frac{1}{q^2}\mathcal{N}.$$

Here J is the Markov operator of the complete graph on $\mathbb{F}^m \setminus \{x\}$. Thus $\lambda(G_5)^2 \approx \frac{1}{q^2}$.

6. Consider a two-step random walk in G_6, $x \to \ell \to y$. If we sample a random line through x then conditioned on $y \neq x$, y is uniformly distributed in $\mathbb{F}^m$. Thus, we can write the Markov operator T associated with this process as:

$$T = \frac{1}{q}I + \left(1 - \frac{1}{q}\right) T',$$

where T' is a Markov operator associated with a random walk on a complete graph on A, without self loops and I is an identity matrix. As $T' = \frac{1}{|A|-1}J - \frac{1}{|A|-1}I$, $\lambda(T') = \frac{1}{q^3-1}$. Thus, $\left|\lambda(T) - \frac{1}{q}\right| \leq \frac{1}{q^3-1}$. The claim follows as $\lambda(G_6)^2 = \lambda(T)$. ◀

Next, we prove Lemma 30. Recall that $\mathcal{A}^s$ denotes set of all s dimensional affine subspaces in $\mathbb{F}^m$. Also, for $r < s$ and for $R \in \mathcal{A}^r$, $\mathcal{A}^s_R \subseteq \mathcal{A}^s$ denotes all those subspaces in $\mathcal{A}^s$ which contains a particular subspace R.

▶ **Lemma 37** (Restatement of Lemma 30). *Let $r \leq k < s \leq \frac{m}{2}$ be integers, and let G be the inclusion graph $G = G(\mathcal{A}^k_R, \mathcal{A}^s_R)$ for an r dimensional subspace R, where $R \neq \emptyset$. Then,*

$$\lambda(G)^2 \leq (1 + o(1)) \cdot q^{-(s-2k+r+1)}.$$

Proof. Fix an r dimensional subspace $R \subseteq \mathbb{F}^m, R \neq \emptyset$ and recall that

$$\mathcal{A}^k_R = \{K \subset \mathbb{F}^m | \dim(K) = k, R \subset K\}.$$

Let $G = G(\mathcal{A}^k_R, \mathcal{A}^s_R)$ be the biregular bipartite inclusion graph and let d_k (resp. d_s) denote the degree of vertex in $\mathcal{A}^k_R$ (resp. $\mathcal{A}^s_R$).

For every $n, t, j \in \mathbb{N}$, let $h(n,t,j)$ be the number of t dimensional subspaces in $\mathbb{F}^n$ that contain a specific dimention j subspace,

$$h(n,t,j) = \frac{(q^n - q^j)\cdots(q^n - q^{t-1})}{(q^t - q^j)\cdots(q^t - q^{t-1})} \approx q^{(n-t)(t-j)}, \tag{11}$$

where $\approx$ denotes equality up to a multiplicative factor $(1 \pm o(1))$, as before. For any fixed j dimensional subspace X, the numerator equals the number of $t - j$ linearly independent points $y_1, y_2, \ldots, y_{t-j}$ in $\mathbb{F}^n$ such that $\dim(\mathtt{span}(X, y_1, y_2, \ldots, y_{t-j})) = t$, whereas for every t dimensional subspace Z, the denominator equals the double counting of Z, i.e the number

of $t-j$ linearly independent points $y_1, y_2, \ldots, y_{t-j}$ such that $\texttt{span}(X, y_1, y_2, \ldots, y_{t-j}) = Z$. We can now bound the number of vertices and the left and right degree in G.

$$\begin{aligned} \left|\mathcal{A}_R^k\right| &= h(m,k,r), & \left|\mathcal{A}_R^s\right| &= h(m,s,r), \\ d_k &= h(m,s,k), & d_s &= h(s,k,r). \end{aligned}$$

Let T be the two-step Markov operator on the bipartite graph G, starting from $\mathcal{A}_R^k$, we want to calculate the entries of T. Let $K_1, K_2 \in \mathcal{A}_R^k$, by definition $(T)_{K_1,K_2}$ is the probability that a two-step random walk will end at K_2, conditioned on it starting from K_1.

Let $r' = \dim(K_1 \cap K_2) \geq r$, in this notation $\dim(K_1 \cup K_2) = 2k - r'$. Any 2 step random walk from K_1 to K_2 looks like $K_1 \to S' \to K_2$ where S' is an s dimentional subspace containing both K_1 and K_2. The number of such S' is exactly $h(m, s, 2k - r')$. Thus, $(T)_{K_1,K_2}$ equals

$$\begin{aligned} (T)_{K_1,K_2} &= \mathbf{Pr}[\text{R.W ends at } K_2 | \text{ R.W starts at } K_1] \\ &= \frac{h(m,s,2k-r')}{d_k \cdot d_s} = \frac{h(m,s,2k-r')}{h(m,s,k) \cdot h(s,k,r)}. \end{aligned} \tag{12}$$

This probability is the same for every $K_1, K_2 \in \mathcal{A}_R^k$ such that $\dim(K_1 \cap K_2) = r'$, so we can denote this value by $p_{r'} = (T)_{K_1,K_2}$. Notice that $p_{r'} \geq p_r$ for every $r' \geq r$.

Let $G_{r'}$ be the graph with vertex set $\mathcal{A}_R^k$, where K_1, K_2 are connected by an edge if $\dim(K_1 \cap K_2) = r'$. We also denote the 0/1 adjacency matrix of graph $G_{r'}$ by $G_{r'}$. With these notations, the 2 step Markov operator T equals

$$T = \sum_{r'=r}^{k} p_{r'} G_{r'}.$$

Notice that this is not a convex combination, $\sum_{r'} p_{r'} \neq 1$, but rather $p_{r'}$ are the entries of T, and $G_{r'}$ are 0/1 matrices.

Let J be the all 1 matrix, we know that $J = \sum_{r'=r}^{k} G_{r'}$. The first matrix in the sum G_r is the only non sparse matrix, since for every subspace $K_1 \in \mathcal{A}_r^k$, almost all other subspaces intersects with K_1 only in R. Therefore we can write $G_r = J - \sum_{r'=r+1}^{k} G_{r'}$, and get

$$T = p_r J + \sum_{r'=r+1}^{k} (p_{r'} - p_r) G_{r'}.$$

Since T is a Markov operator of a regular graph, the all $\mathbf{1}$ vector is the vector with the maximal eigenvalue, which equals 1. Since $G_{r'}$ are also regular graphs, $\mathbf{1}$ is the vector with the maximal eigenvalue, which equals $\deg(G_{r'})$, which is the number of $K' \in \mathcal{A}_R^k$ such that $\dim(K \cap K') = r'$ (as the adjacency matrices are not normalized).

$$\begin{aligned} \deg(G_{r'}) =& h(k,r',r) \cdot \frac{(q^m - q^k) \cdots (q^m - q^{2k-r'-1})}{(q^k - q^{r'}) \cdots (q^k - q^{k-1})} \\ \approx& q^{(k-r')(r'-r)} \cdot q^{(m-k)(k-r')} = q^{(k-r')(m-k+r'-r)} \end{aligned}$$

For every $K \in \mathcal{A}_R^k$, the factor $h(k, r', r)$ is the number of r' dimensional subspace in K that contain R, the second factor is the number of k dimensional subspaces that intersect with K only in a specific r' dimensional subspace.

Let v be the normalized eigenvector of the second eigenvalue of T, this means that $v \perp \mathbf{1}$ and $\|v\| = 1$. Since J is the all 1 matrix, $Jv = 0$. We also know that for every $r' > r$,

$\|G_{r'}v\| \leq \deg(G_{r'})$, as it is true for every vector v.

$$\begin{aligned}\|Tv\| &= \left\| \sum_{r'=r+1}^{k} (p_{r'} - p_r) G_{r'} v \right\| \\ &\leq \sum_{r'=r+1}^{k} (p_{r'} - p_r) \|G_{r'}v\| && \text{(triangle inequality)} \\ &\leq \sum_{r'=r+1}^{k} p_{r'} \deg(G_{r'})\end{aligned}$$

For every r', by using the expression for $p_{r'}$ from (12) and bounds on h from (11) we get that

$$p_{r'} \deg(G_{r'}) \approx p_{r'} q^{(k-r')(m-s+r'-r)} \approx q^{-(r'-r)(s-2k+r')}.$$

Since $r' > r$, $(r'-r)(s-2k+r')$ is minimized when $r' = r+1$ and hence

$$\lambda(T) = \|Tv\| \leq (1+o(1)) \sum_{r'=r+1}^{k} \frac{1}{q^{s-2k+r'}} \leq (1+o(1)) \cdot \frac{1}{q^{s-2k+r+1}}.$$

The lemma statement now follows from the Fact 35. ◀

B Spectral Expansion Properties Proofs

▶ **Lemma 38** (Restatement of Lemma 6). *Let D_1, D_2 as defined in Definition 5. Let $G = (A \cup B, E)$ be a bi-regular bipartite graph, then for every subset $B' \subset B$ of measure $\mu > 0$ and every $E' \subset E$*

$$\left| \mathop{\mathbf{Pr}}_{(a,b)\sim D_1}[(a,b) \in E'] - \mathop{\mathbf{Pr}}_{(a,b)\sim D_2}[(a,b) \in E'] \right| \leq \frac{\lambda(G)}{\sqrt{\mu}}.$$

Where is D_2 returned $\perp$, we treat is as it is not in E'.

Proof. In the proof we represent both probabilities as an inner product, and then use $\lambda(G)$ to bound the difference. Let $M \in \mathbb{R}^{A \times B}$ the adjacency matrix of the graph G, normalized such that $M\mathbf{1} = \mathbf{1}$ (where the first $\mathbf{1}$ is of dimension $|B|$ and the second of dimension $|A|$). We define the matrix M' representing the subset of edges E', $M'_{a,b} = M_{a,b} \cdot (\mathbf{1}_{E'})_{a,b}$.

Starting with the probability of $(a,b) \sim D_1$, the vector $M'\mathbf{1}_{B'}$ satisfies that for every $a \in A$, $(M'\mathbf{1}_{B'})_a = \mathbf{Pr}_{b\in N(a)}[(a,b) \in E', b \in B']$.

$$\begin{aligned}\langle \mathbf{1}, M'\mathbf{1}_{B'} \rangle &= \mathop{\mathbf{E}}_{a\sim A}\left[E_{b\sim N(a)}[\mathbb{I}((a,b) \in E', b \in B')]\right] \\ &= \mathop{\mathbf{Pr}}_{a\sim A, b\sim N(a)}[(a,b) \in E', b \in B'] && \text{(using bi-regularity of } G) \\ &= \mathop{\mathbf{Pr}}_{b\sim B, a\sim N(b)}[(a,b) \in E', b \in B'] \\ &= \mathop{\mathbf{Pr}}_{b\sim B}[b \in B'] \cdot \mathop{\mathbf{Pr}}_{b\sim B, a\sim N(b)}[(a,b) \in E' \mid b \in B'] \\ &= \mu \cdot \mathop{\mathbf{Pr}}_{(a,b)\sim D_1}[(a,b) \in E'].\end{aligned}$$

We now want to represent the second probability as an inner product. We define the vector $P \in [0,1]^A$ as follows, for each $a \in A$:

1. If $N(a) \cap B' = \emptyset$, then $P_a = 0$.
2. Else, $P_a = \mathbf{Pr}_{b \in N(a)}[(a,b) \in E' \mid b \in B']$.

In this notation $\mathbf{Pr}_{(a,b)\sim D_2}[(a,b) \in E'] = \langle \mathbf{1}, P \rangle$.

We now want to find a connection between the inner products. If $P_a \neq 0$, then it defined as the conditional probability, and

$$\mathop{\mathbf{Pr}}_{b\sim N(a)}[b \in B', (a,b) \in E'] = \mathop{\mathbf{Pr}}_{b\sim N(a)}[b \in B'] \mathop{\mathbf{Pr}}_{b\sim N(a)}[(a,b) \in E' \mid b \in B'] = \mathop{\mathbf{Pr}}_{b\sim N(a)}[b \in B'] P_a.$$

If $P_a = 0$ then also $\mathbf{Pr}_{b\sim N(a)}[b \in B', (a,b) \in E'] = 0$, and the above equality still holds. We notice that $(M'\mathbf{1}_{B'})_a = \mathbf{Pr}_{b\in N(a)}[(a,b) \in E', b \in B']$ and $(M\mathbf{1}_{B'})_a = \mathbf{Pr}_{b\in N(a)}[b \in B']$, which means that for every $a \in A$, $(M'\mathbf{1}_{B'})_a = (M\mathbf{1}_{B'})_a P_a$ and

$$\langle M\mathbf{1}_{B'}, P \rangle = \langle \mathbf{1}, M'\mathbf{1}_{B'} \rangle.$$

Therefore we can express the difference between the two probabilities as

$$\begin{aligned}
\left| \mathop{\mathbf{Pr}}_{(a,b)\sim D_1}[(a,b) \in E'] - \mathop{\mathbf{Pr}}_{(a,b)\sim D_2}[(a,b) \in E'] \right| &= \left| \frac{1}{\mu} \langle \mathbf{1}, M'\mathbf{1}_{B'} \rangle - \langle \mathbf{1}, P \rangle \right| \\
&= \left| \frac{1}{\mu} \langle M\mathbf{1}_{B'}, P \rangle - \langle \mathbf{1}, P \rangle \right| \\
&= \frac{1}{\mu} \left| \langle M\mathbf{1}_{B'} - \mu\mathbf{1}, P \rangle \right| \\
&\leq \frac{1}{\mu} \left\| M\mathbf{1}_{B'} - \mu\mathbf{1} \right\| \left\| P \right\| \quad \text{(By Cauchy Swartz)}
\end{aligned} \tag{13}$$

Since P is a vector in $[0,1]$ and the inner product we use is expectation, $\|P\| \leq 1$. In order to finish the proof we need to bound the size of the vector

$$M\mathbf{1}_{B'} - \mu\mathbf{1} = M\mathbf{1}_{B'} - \mu M\mathbf{1} = M(\mathbf{1}_{B'} - \mu\mathbf{1}).$$

We notice that $\mathbf{1}_{B'}$ is a $\{0,1\}$ vector of measure μ, so $\langle \mathbf{1}_{B'}, \mathbf{1} \rangle = \langle \mathbf{1}_{B'}, \mathbf{1}_{B'} \rangle = \mu$, and $(\mathbf{1}_{B'} - \mu\mathbf{1}) \perp \mathbf{1}_B$. By the definition of $\lambda(G)$, this means that

$$\|M(\mathbf{1}_{B'} - \mu\mathbf{1})\| \leq \lambda(G) \|\mathbf{1}_{B'} - \mu\mathbf{1}\| \leq \lambda\sqrt{\mu}.$$

We substitute the norm of the vector in equation (13) and we are done. ◀

▶ **Lemma 39** (Restatement of Lemma 8). *Let D_3, D_4 as defined in Definition 7. Let $G = (A \cup B, E)$ be a bi-regular bipartite graph, such that every two distinct $b_1, b_2 \in B$ have exactly the same number of common neighbors (i.e for all distinct $b_1, b_2 \in B$, $|N(b_1) \cap N(b_2)|$ is the same), and this number is non-zero. Then for every subset $B' \subset B$ of measure $\mu > 0$ and every $E' \subset E$*

$$\left| \mathop{\mathbf{Pr}}_{a,b_1,b_2\sim D_3}[(a,b_1)(a,b_2) \in E'] - \mathop{\mathbf{Pr}}_{a,b_1,b_2\sim D_4}[(a,b_1)(a,b_2) \in E'] \right| \leq \frac{2\lambda(G)}{\mu} + \frac{1}{\mu^2 d_A} + \frac{1}{\mu^2 |B|}$$

Where is D_4 returned $\perp$, we treat is as it is not in E' and d_A is the degree on A side.

Proof. This proof is similar in spirit to the proof of Lemma 6, with more complication since the event contains two edges instead of a single one.

Let $M \in \mathbb{R}^{A \times B}$ the adjacency matrix of the graph G, normalized such that $M\mathbf{1} = \mathbf{1}$. We denote by M' the matrix that represents the edges in E', i.e for each $a \in A, b \in B$, $M'_{a,b} = M_{a,b} \cdot (\mathbf{1}_{E'})_{a,b}$.

Starting from D_3, we first write the conditional probability

$$\begin{aligned}\Pr_{\substack{b_1,b_2\\ a\sim N(b_1)\cap N(b_2)}}&[b_1,b_2\in B',(a,b_1),(a,b_2)\in E']\\ &=\Pr_{b_1,b_2}[b_1,b_2\in B']\Pr_{a,b_1,b_2\sim D_3}[(a,b_1),(a,b_2)\in E']\\ &=\mu^2\Pr_{a,b_1,b_2\sim D_3}[(a,b_1),(a,b_2)\in E'].\end{aligned}\tag{14}$$

We want to express the left side as an inner product, we notice that for each $a\in A$:

$$(M'\mathbf{1}_{B'})_a=\mathop{\mathbf{E}}_{b\sim N(a)}[\mathbb{I}(b\in B',(a,b)\in E')].$$

Therefore the inner product satisfies

$$\begin{aligned}\langle M'\mathbf{1}_{B'},M'\mathbf{1}_{B'}\rangle&=\mathop{\mathbf{E}}_{a\sim A}\left[\mathop{\mathbf{E}}_{b_1,b_2\sim N(a)}[\mathbb{I}(b_1,b_2\in B',(a,b_1)(a,b_2)\in E')]\right]\\ &=\Pr_{a\sim A,b_1,b_2\sim N(a)}[b_1,b_2\in B',(a,b_1)(a,b_2)\in E']\end{aligned}\tag{15}$$

Since each two $b_1,b_2\in B$ has the same number of neighbors,

$$\Pr_{\substack{a\sim A\\ b_1\neq b_2\sim N(a)}}[b_1,b_2\in B',(a,b_1)(a,b_2)\in E']=\Pr_{\substack{b_1\neq b_2\sim B\\ a\sim N(b_1)\cap N(b_2)}}[b_1,b_2\in B',(a,b_1)(a,b_2)\in E'].$$

We want to switch the expression in (15) by the one is (14), we know that they are equal when $b_1\neq b_2$. But the probability of $b_1=b_2$ is different between the two cases, it is $\frac{1}{d_A}$ if we pick neighbors of a and $\frac{1}{|B|}$ if we pick two random vertices in B. If we add the probability of $b_1=b_2$ as an error, we get that

$$\left|\mu^2\Pr_{a,b_1,b_2\sim D_3}[(a,b_1)(a,b_2)\in E']-\langle M'\mathbf{1}_{B'},M'\mathbf{1}_{B'}\rangle\right|\leq\frac{1}{d_A}+\frac{1}{|B|}\tag{16}$$

Now we want to express the probability of $a,b_1,b_2\sim D_4$ as an inner product. In order to do that, we define the vector P, for every $a\in A$

1. If $N(a)\cap B'=\emptyset$, then $P_a=0$.
2. Else, $P_a=\mathbf{Pr}_{b_1,b_2\sim N(a)}[(a,b_1)(a,b_2)\in E'\mid b_1,b_2\in B']$.

The vector P is defined such that

$$\Pr_{a,b_1,b_2\sim D_4}[(a,b_1)(a,b_2)\in E']=\mathop{\mathbf{E}}_{a}[P_a]=\langle\mathbf{1},P\rangle.$$

We want to find a connection between this expression and the expression representing the probability $\mathbf{Pr}_{a,b_1,b_2\sim D_3}[(a,b_1)(a,b_2)\in E']$.

We use (16) and the triangle inequality to bound the difference between the two target probabilities

$$\begin{aligned}&\left|\Pr_{a,b_1,b_2\sim D_3}[(a,b_1)(a,b_2)\in E']-\Pr_{a,b_1,b_2\sim D_4}[(a,b_1)(a,b_2)\in E']\right|\\ &\quad\leq\left|\frac{1}{\mu^2}\langle M'\mathbf{1}_{B'},M'\mathbf{1}_{B'}\rangle-\langle\mathbf{1},P\rangle\right|+\frac{1}{\mu^2 d_A}+\frac{1}{\mu^2|B|}\end{aligned}\tag{17}$$

We now need to bound the expression in (17), in order to do that, we will first show that

$$\langle M'\mathbf{1}_{B'},M'\mathbf{1}_{B'}\rangle=\Pr_{a\sim A,b_1,b_2\sim N(a)}[(a_1,b)(a_2,b)\in E',b_1,b_2\in B']=\mathop{\mathbf{E}}_{a}[P_a(M\mathbf{1}_{B'})_a^2].\tag{18}$$

We notice that for a such that $P_a > 0$, it equals the conditional probability and

$$\mathop{\mathbf{Pr}}_{b_1,b_2\sim N(a)}[(a_1,b)(a_2,b)\in E', b_1,b_2\in B'] = \mathop{\mathbf{Pr}}_{b_1,b_2\sim N(a)}[b_1,b_2\in B']P_a.$$

If a is such that $P_a = 0$, then $\mathbf{Pr}_{b_1,b_2\sim N(a)}[(a_1,b)(a_2,b)\in E', b_1,b_2\in B'] = 0$ and the above equality still holds. We further notice that

$$(M\mathbf{1}_{B'})_a = \mathop{\mathbf{E}}_{b\sim N(a)}[\mathbb{I}(b\in B')].$$

If we substitute $\mathbf{Pr}_{b_1,b_2\sim N(a)}[b_1,b_2\in B']$ in $(M\mathbf{1}_{B'})_a^2$, we get (18).

In order to finish the proof, we upper bound

$$\left|\frac{1}{\mu^2}\langle M'\mathbf{1}_{B'}, M'\mathbf{1}_{B'}\rangle - \langle \mathbf{1}, P\rangle\right| = \left|\mathop{\mathbf{E}}_a\left[\frac{1}{\mu^2}P_a(M\mathbf{1}_{B'})_a^2 - P_a\right]\right| = \frac{1}{\mu^2}\left|\mathop{\mathbf{E}}_a[P_a((M\mathbf{1}_{B'})_a^2-\mu^2)]\right|.$$

We now upper bound the expectation as follows,

$$\begin{aligned}
\mathop{\mathbf{E}}_a[P_a((M\mathbf{1}_{B'})_a^2-\mu^2)] &= \mathop{\mathbf{E}}_a[P_a((M\mathbf{1}_{B'})_a-\mu)((M\mathbf{1}_{B'})_a+\mu)] \\
&\le \max_a\{|P_a|\}\mathop{\mathbf{E}}_a[|((M\mathbf{1}_{B'})_a-\mu)((M\mathbf{1}_{B'})_a+\mu)|] \\
&\le \|M\mathbf{1}_{B'}-\mu\mathbf{1}\|\,\|M\mathbf{1}_{B'}+\mu\mathbf{1}\| \qquad (19)\\
&\le \lambda\sqrt{\mu}\sqrt{4\mu}, \qquad (20)
\end{aligned}$$

where (19) is due to Cauchy-Schwarz inequality and using $|P_a|\le 1$. In (20), we bound $\|M\mathbf{1}_{B'}-\mu\mathbf{1}\|$ like in the previous proof,

$$\|M\mathbf{1}_{B'}-\mu\mathbf{1}\| = \|M\mathbf{1}_{B'}-\mu M\mathbf{1}\| = \|M(\mathbf{1}_{B'}-\mu\mathbf{1})\| \le \lambda\|\mathbf{1}_{B'}\| \le \lambda\sqrt{\mu}.$$

Finally, we bound $\|M\mathbf{1}_{B'}+\mu\mathbf{1}\|$:

$$\begin{aligned}
\|M\mathbf{1}_{B'}+\mu\mathbf{1}\|^2 &= \langle M\mathbf{1}_{B'}+\mu\mathbf{1}, M\mathbf{1}_{B'}+\mu\mathbf{1}\rangle \\
&= \langle M\mathbf{1}_{B'}, M\mathbf{1}_{B'}\rangle + 2\langle M\mathbf{1}_{B'},\mu\mathbf{1}\rangle + \langle\mu\mathbf{1},\mu\mathbf{1}\rangle \\
&\le \|\mathbf{1}_{B'}\|^2 + 2\mu + \mu^2\|\mathbf{1}\|^2 \\
&\le \mu + 2\mu + \mu^2 \le 4\mu.
\end{aligned}$$

◀

C Rubinfeld-Sudan Characterization

In this section, we present a proof of Theorem 21. The proof uses the following fact from [13]:

▶ **Fact 40.** *Let $f:\mathbb{F}^m\to\mathbb{F}$ be a function, and let $N_{y,h} = \{y+ih \mid i\in\{0,\dots,d+1\}\}$. f is degree d iff it satisfies the following identity for all y and h:*

$$\sum_{i=0}^{d+1}\alpha_i f(y+ih) = 0,$$

where $\alpha_i = \binom{d+1}{i}(-1)^{i+1}$.

Throughout this section we let $\alpha_i = \binom{d+1}{i}(-1)^{i+1}$ as in the above fact.

▶ **Theorem 41** (Restatement of Theorem 21). *Let $f : \mathbb{F}^m \to \mathbb{F}$ be a function, and let $N_{y,h} = \{y + ih \mid i \in \{0, \ldots, d+1\}\}$, if f satisfies*

$$\Pr_{y,h \in \mathbb{F}^m}[\exists \deg d \text{ polynomial } p \textbf{ s.t. } p_{|_{N_{y,h}}} = f_{|_{N_{y,h}}}] \geq 1 - \delta, \tag{21}$$

for $\delta \leq \frac{1}{2(d+2)^2}$, then there exists a degree d polynomial g such that $f \overset{2\delta}{\approx} g$.

Proof. Define a function $g : \mathbb{F}^m \to \mathbb{F}$ to be $g(y) = \mathtt{maj}_{h \in \mathbb{F}^m}\{\sum_{i=1}^{d+1} \alpha_i f(y + ih)\}$ breaking the ties arbitrarily. Next we argue that g is very close to f and g itself is a degree d function.

To see that g is $(1 - 2\delta)$ close to f, consider the set of all y for which $\mathbf{Pr}_h[f(y) = \sum_{i=1}^{d+1} \alpha_i f(y + ih)] > 1/2$. For all these y, $f(y) = g(y)$ as g was the majority vote. It is easy to see that fraction of y for which the probability is at most $1/2$ is at most 2δ as otherwise it will contradict the hypothesis (21). The rest of the proof will be proving the following two claims.

▶ **Claim 42.** *For all $y \in \mathbb{F}^m$, $\mathbf{Pr}_h[g(y) = \sum_{i=1}^{d+1} \alpha_i f(y + ih)] \geq 1 - 2(d+1)\delta$.*

▶ **Claim 43.** *For all y and h in $\mathbb{F}^m$, we have $\sum_{i=0}^{d+1} \alpha_i g(y + ih) = 0$.*

Claim 43 and Fact 40 imply that g is in fact a degree d function and hence the theorem follows. We now proceed with proving these two claims.

Proof of Claim 42: We will show that for all $y \in \mathbb{F}^m$,

$$\Pr_{h_1,h_2}\left[\sum_{i=1}^{d+1} \alpha_i f(y + ih_1) = \sum_{j=1}^{d+1} \alpha_j f(y + jh_2)\right] \geq 1 - 2(d+1)\delta. \tag{22}$$

Note that this is enough to prove the claim. To see this, let $p_a = \mathbf{Pr}_h[\sum_{i=1}^{d+1} \alpha_i f(y + ih) = a]$ for $a \in \mathbb{F}$. Then (22) becomes $\sum_{a \in \mathbb{F}} p_a^2 \geq 1 - 2(d+1)\delta$. Since $g(y)$ was the majority vote, we have $\mathbf{Pr}_h[g(y) = \sum_{i=1}^{d+1} \alpha_i f(y + ih)] = \max_{a \in \mathbb{F}} p_a \geq \sum_{a \in \mathbb{F}} p_a^2 \geq 1 - 2(d+1)\delta$.

To prove (22), consider the following $(d+2) \times (d+2)$ matrix Z with $(i,j)^{th}$ entry $Z_{i,j} = \alpha_i \alpha_j f(y + ih_1 + jh_2)$, for $i, j \in \{0, \ldots, d+1\}$.

$$Z = \begin{bmatrix} f(y) & \ldots & \alpha_0\alpha_j f(y + jh_2) & \ldots \\ \vdots & \ddots & \vdots & \ddots \\ \alpha_i\alpha_0 f(y + ih_1) & \ldots & \alpha_i\alpha_j f(y + ih_1 + jh_2) & \ldots \\ \vdots & \ddots & \vdots & \ddots \end{bmatrix}$$

If $h_1 \in \mathbb{F}^m$ u.a.r then for any $i \in \{1, 2, \ldots, d+1\}$, ih_1 is distributed uniformly in $\mathbb{F}^m$. Same is true for h_2 and jh_2. Consider the following events:

- For every $i \in \{1, 2, \ldots, d+1\}$, R_i be the event that the sum of the i'th row is *zero*, i.e $\sum_{j=0}^{d+1} Z_{i,j} = 0$.
- For every $j \in \{1, 2, \ldots, d+1\}$, C_j be the event that sum of the j'th column is *zero*, i.e $\sum_{i=0}^{d+1} Z_{i,j} = 0$.

Note that R_i, C_j are not defined for the first row and column ($i = 0$ and $j = 0$). Using the hypothesis (21) of the theorem and Fact 40, we have

$$\Pr_{h_1,h_2}[R_i] \geq 1 - \delta, \qquad \forall i \in \{1, 2, \ldots, d+1\}$$

$$\Pr_{h_1,h_2}[C_j] \geq 1 - \delta, \qquad \forall j \in \{1, 2, \ldots, d+1\}$$

The event in (22) is same as $\sum_{i=1}^{d+1} Z_{i,0} = \sum_{j=1}^{d+1} Z_{0,j}$ (note that the sums don't include the first element, $Z_{0,0}$). If all the above events R_i, C_j happen then $\sum_{i=1}^{d+1} Z_{i,0} = \sum_{j=1}^{d+1} Z_{0,j} = -\sum_{i,j=1}^{d+1} Z_{i,j}$. By using union bound we get $\mathbf{Pr}[\wedge_{i=1}^{d+1} R_i \wedge_{j=1}^{d+1} C_j] \geq 1 - 2(d+1)\delta$ which implies (22).

Proof of Claim 43: In this case, consider the following $(d+2) \times (d+2)$ matrix Y whose $(i,j)^{th}$ entry is $Y_{i,j} = \alpha_i \alpha_j f(y + ih + j(h_1 + ih_2))$ except when $j = 0$. When $j = 0$, $Y_{i,0} = \alpha_i \alpha_0 g(y + ih)$.

$$Y = \begin{bmatrix} \alpha_0\alpha_0 g(y) & \dots & \alpha_0\alpha_j f(y + jh_1) & \dots \\ \vdots & \ddots & \vdots & \ddots \\ \alpha_i\alpha_0 g(y + ih) & \dots & \alpha_i\alpha_j f(y + ih + j(h_1 + ih_2)) & \dots \\ \vdots & \ddots & \vdots & \ddots \end{bmatrix}$$

Define the following set of events:

- For $i \in \{0, 1, \dots, d+1\}$, R_i be the event that the sum of all elements from row i is *zero*, i.e $\sum_{i=0}^{d+1} Y_{i,j} = 0$.
- For $j \in \{0, 1, \dots, d+1\}$, C_j be the event that the sum of all elements from column j is *zero*, i.e $\sum_{j=0}^{d+1} Y_{i,j} = 0$.

Let h_1, h_2 are independent and distributed u.a.r in $\mathbb{F}^m$. As the event C_0 is independent of h_1 and h_2, in order to prove the claim it is enough to show that $\mathbf{Pr}_{h_1,h_2}[C_0] > 0$.

For each row $i \in \{0, 1, 2, \dots, d+1\}$ we apply Claim 42 with $y' = y + ih$ and $h' = h_1 + ih_2$, and get $\mathbf{Pr}_{h_1,h_2}[\neg R_i] \leq 2(d+1)\delta$ (note that $\alpha_0 = -1$). If h_1, h_2 are independent and distributed u.a.r in $\mathbb{F}^m$ then so are $(y + jh_1)$ and $(h + h_2)$. Therefore, using the hypothesis (21) of the theorem and Fact 40, we have for all columns except $j = 0$, $\mathbf{Pr}_{h_1,h_2}[\neg C_j] \leq \delta$. Using union bound, we get

$$\mathop{\mathbf{Pr}}_{h_1,h_2}\left[\bigwedge_{i=0}^{d+1} R_i \bigwedge_{j=1}^{d+1} C_j\right] \geq 1 - 2(d+1)(d+2)\delta + (d+1)\delta > 0.$$

The claim now follows using the observation that the event C_0 is implied by the event $\wedge_{i=0}^{d+1} R_i \wedge_{j=1}^{d+1} C_j$. To see this, the event $\wedge_{i=0}^{d+1} R_i$ implies that the sum of all entries in Y is *zero* whereas $\wedge_{j=1}^{d+1} C_j$ implies that the sum of all elements from the submatrix $(Y_{i,j})_{j=1}^{d+1}$ is *zero*. Hence, if both these events happen then the sum of all elements from column 0 must be *zero*. ◀

On the Power of Learning from k-Wise Queries

Vitaly Feldman[1] **and Badih Ghazi**[*2]

1 **IBM Research - Almaden, USA**
vitaly@post.harvard.edu
2 **Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, USA**
badih@mit.edu

Abstract

Several well-studied models of access to data samples, including statistical queries, local differential privacy and low-communication algorithms rely on queries that provide information about a function of a single sample. (For example, a statistical query (SQ) gives an estimate of $\mathbb{E}_{x\sim D}[q(x)]$ for any choice of the query function $q : X \to \mathbb{R}$, where D is an unknown data distribution.) Yet some data analysis algorithms rely on properties of functions that depend on multiple samples. Such algorithms would be naturally implemented using k-wise queries each of which is specified by a function $q : X^k \to \mathbb{R}$. Hence it is natural to ask whether algorithms using k-wise queries can solve learning problems more efficiently and by how much.

Blum, Kalai, Wasserman [9] showed that for any weak PAC learning problem over a fixed distribution, the complexity of learning with k-wise SQs is smaller than the (unary) SQ complexity by a factor of at most 2^k. We show that for more general problems over distributions the picture is substantially richer. For every k, the complexity of distribution-independent PAC learning with k-wise queries can be exponentially larger than learning with $(k+1)$-wise queries. We then give two approaches for simulating a k-wise query using unary queries. The first approach exploits the structure of the problem that needs to be solved. It generalizes and strengthens (exponentially) the results of Blum et al. [9]. It allows us to derive strong lower bounds for learning DNF formulas and stochastic constraint satisfaction problems that hold against algorithms using k-wise queries. The second approach exploits the k-party communication complexity of the k-wise query function.

1998 ACM Subject Classification I.2.6 Learning

Keywords and phrases Statistical Queries, PAC Learning, Differential Privacy, Lower bounds, Communication Complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.41

1 Introduction

In this paper, we consider several well-studied models of learning from i.i.d. samples that restrict the algorithm's access to samples to evaluation of functions of an individual sample. The primary model of interest is the statistical query model introduced by Kearns [31] as a restriction of Valiant's PAC learning model [39]. The SQ model allows the learning algorithm to access the data only via *statistical queries*, which are estimates of the expectation of any function of labeled examples with respect to the input distribution D. More precisely, if the domain of the functions is Z, then a statistical query is specified by a function

* Part of this work was done while the author was at IBM Research - Almaden. The author is supported in part by NSF STC Award CCF 0939370 and NSF Award CCF-1217423.


© Vitaly Feldman and Badih Ghazi;
licensed under Creative Commons License CC-BY

$\phi : Z \times \{\pm 1\} \to [-1, 1]$ and by a tolerance parameter τ. Given ϕ and τ, the statistical query oracle returns a value v which satisfies $|v - \mathbb{E}_{(z,b)\sim D}[\phi(z, b)]| \leq \tau$.

The SQ model is known to be closely-related to several other models and concepts: linear statistical functionals [41], learning with a distance oracle [5], approximate counting (or linear) queries extensively studied in differential privacy (e.g., [16, 7, 20, 34]), local differential privacy [30], evolvability [40, 23], and algorithms that extract a small amount of information from each sample [4, 26, 28, 36]. This allows to easily extend the discussion in the context of the SQ model to these related models and we will formally state several such corollaries.

Most standard algorithmic approaches used in learning theory are known to be implementable using SQs (e.g., [8, 17, 7, 11, 28, 3, 27]) leading to numerous theoretical (e.g., [2, 14, 19]) and practical (e.g., [11, 35, 38, 18]) applications. SQ algorithms have also been recently studied outside the context of learning theory [26, 28, 27]. In this case we denote the domain of data samples by X.

Another reason for the study of SQ algorithms is that it is possible to prove information-theoretic lower bounds on the complexity of any SQ algorithm that solves a given problem. Given that a large number of algorithmic approaches to problems defined over data sampled i.i.d. from some distribution can be implemented using statistical queries, this provides a strong and unconditional evidence of the problem's hardness. For a number of central problems in learning theory and complexity theory, unconditional lower bounds for SQ algorithms are known that closely match the known *computational* complexity upper bounds for those problems (*e.g.* [6, 26, 28, 12, 15]).

A natural strengthening of the SQ model (and other related models) is to allow function over k-tuples of samples instead of a single sample. That is, for a k-ary query function $\phi : X^k \to [-1, 1]$, the algorithm can obtain an estimate of $\mathbb{E}_{x_1,\ldots,x_k\sim D}[\phi(x_1, \ldots, x_k)]$. It can be seen as interpolating between the power of algorithms that can see all the samples at once and those that process a single sample at a time. While most algorithms can be implemented using standard unary queries, some algorithms are known to require such more powerful queries. The most well-known example is Gaussian elimination over $\mathbb{F}_2^n$ that is used for learning parity functions. Standard hardness amplification techniques rely on mapping examples of a function $f(z)$ to examples of a function $g(f(z_1), \ldots, f(z_k))$ (for example [10, 22]). Implementing such reduction requires k-wise queries and, consequently, to obtain a lower bound for solving an amplified problem with unary queries one needs a lower bound against solving the original problem with k-wise queries. A simple example of 2-wise statistical query is collision probability $\Pr_{x_1,x_2\sim D}[x_1 = x_2]$ that is used in several distribution property testing algorithms.

1.1 Previous work

Blum, Kalai and Wasserman [9] introduced and studied the power of k-wise SQs in the context of weak *distribution-specific* PAC learning: that is the learning algorithm observes pairs (z, b), where z is chosen randomly from some fixed and known distribution P over Z and $b = f(z)$ for some unknown function f from a class of functions $\mathcal{C}$. They showed that if a class of functions $\mathcal{C}$ can be learned with error $1/2 - \lambda$ relative to distribution P using q k-wise SQs of tolerance τ then it can be learned with error $\max\{1/2 - \lambda, 1/2 - \tau/2^k\}$ using $O(q \cdot 2^k)$ unary SQs of tolerance $\tau/2^k$.

More recently, Steinhardt et al. [36] considered k-wise queries in the b-bit sampling model in which for any query function $\phi : X^k \to \{0, 1\}^b$ an algorithm get the value $\phi(x_1, \ldots, x_k)$ for $x_1, \ldots, x_k$ drawn randomly and independently from D (it is referred to as one-way communication model in their work). They give a general technique for proving lower bounds on the number of such queries that are required to solve a given problem.

1.2 Our results

In this work, we study the relationship between the power of k-wise queries and unary queries for arbitrary problems in which the input is determined by some unknown input distribution D that belongs a (known) family of distributions $\mathcal{D}$ over domain X.

1.2.1 Separation for distribution-independent learning

We first demonstrate that for distribution-independent PAC learning $(k+1)$-wise queries are exponentially stronger than k-wise queries. We say that the k-wise SQ complexity of a certain problem is m if m is the smallest such that there exists an algorithm that solves the problem using m k-wise SQs of tolerance $1/m$.

▶ **Theorem 1.** *(Informal) For every positive integer k and any prime number p, there is a concept class $\mathcal{C}$ of Boolean functions defined over a domain of size p^{k+1} such that the $(k+1)$-wise SQ complexity of distribution-independent PAC learning $\mathcal{C}$ with is $O_k(\log p)$ whereas the k-wise SQ complexity of distribution-independent PAC learning of $\mathcal{C}$ is $\Omega_k(p^{1/4})$.*

The class of functions we use consists of all indicator functions of k-dimensional affine subspaces of $\mathbb{F}_p^{k+1}$. Our lower bound is a generalization of the lower bound for unary SQs in [25] (that corresponds to $k=1$ case of the lower bound). A simple but important observation that allows us to easily adapt the techniques from earlier works on SQs to the k-wise case is that a k-wise SQ for an input distribution $D \in \mathcal{D}$ are equivalent to unary SQ for a product distribution D^k.

The upper bound relies on the ability to find the affine subspace given $k+1$ positively labeled and linearly independent points in $\mathbb{F}_p^{k+1}$. Unfortunately, for general distributions the probability of observing such a set of points can be arbitrarily small. Nevertheless, we argue that there will exist a unique lower-dimensional affine subspace that contains enough probability mass of all the positive points in this case. This upper bound essentially implies that given k-wise queries one can solve problems that require Gaussian elimination over a system of k equations.

1.2.2 Reduction for flat $\mathcal{D}$

The separation in Theorem 1 relies on using an unrestricted class of distributions $\mathcal{D}$. We now prove that if $\mathcal{D}$ is "flat" relative to some "central" distribution $\bar{D}$ then one can upper bound the power of k-wise queries in terms of unary queries.

▶ **Definition 1.1** (Flat class of distributions)**.** *Let $\mathcal{D}$ be a set of distributions over X, and $\bar{D}$ a distribution over X. For $\gamma \geq 1$ we say that $\mathcal{D}$ is γ-flat if there exists some distribution $\bar{D}$ over X such that for all $D \in \mathcal{D}$ and all measurable subsets $E \subseteq X$, we have that $\Pr_{x\sim D}[x \in E] \leq \gamma \cdot \Pr_{x\sim \bar{D}}[x \in E]$.*

We now state our upper bound for flat classes of distributions, where we use $\mathsf{STAT}_D^{(k)}(\tau)$ to refer to the oracle that answers k-wise SQs for D with tolerance τ.

▶ **Theorem 2.** *Let $\gamma \geq 1$, $\tau > 0$ and k be any positive integer. Let X be a domain and $\mathcal{D}$ a γ-flat class of distributions over X. There exists a randomized algorithm that given any $\delta > 0$ and a k-ary function $\phi : X^k \to [-1,1]$ estimates $D^k[\phi]$ within τ for every (unknown) $D \in \mathcal{D}$ with success probability at least $1-\delta$ using*

$$\tilde{O}\left(\frac{\gamma^{k-1}\cdot k^3}{\tau^3}\cdot \log(1/\delta)\right)$$

queries to $\mathsf{STAT}_D^{(1)}(\tau/(6 \cdot k))$.

To prove this result, we use a recent general characterization of SQ complexity [25]. This characterization reduces the problem of estimating $D^k[\phi]$ to the problem of distinguishing between D^k and D_1^k for every $D \in \mathcal{D}$ and some fixed D_1. We show that when solving this problem, any k-wise query can be replaced by a randomly chosen set of unary queries. Finding these queries requires drawing samples from D^{k-1}. As we do not know D, we use $\bar{D}$ instead incurring the γ^{k-1} overhead in sampling. In Section 4 we show that weaker notions of "flatness" based on different notions of divergence between distributions can also be used in this reduction.

It is easy to see that, when PAC learning $\mathcal{C}$ with respect to a fixed distribution P over Z, the set of input distributions is 2-flat (relative to the distribution that is equal to P on Z and gives equal weight $1/2$ to each label). Therefore, our result generalizes the results in [9]. More importantly, the tolerance in our upper bound scales linearly with k rather than exponentially (namely, $\tau/2^k$).

This result can be used to obtain lower bounds against k-wise SQs algorithms from lower bounds against unary SQ algorithms. In particular, it can be used to rule out reductions that require looking at k points of the original problem instance to obtain each point of the new problem instance. As an application, we obtain exponential lower bounds for solving constraint stochastic satisfaction problems and DNF learning by k-wise SQ algorithm with $k = n^{1-\alpha}$ for any constant $\alpha > 0$ from lower bounds for CSPs given in [28]. We state the result for learning DNF here. Definitions and the lower bound for CSPs can be found in Section 4.3.

▶ **Theorem 3.** *For any constant $\alpha > 0$ (independent of n), there exists a constant $\beta > 0$ such that any algorithm that learns DNF formulas of size n with error $< 1/2 - n^{-\beta \log n}$ and success probability at least $2/3$ requires at least $2^{n^{1-\alpha}}$ calls to $\mathsf{STAT}_D^{(n^{1-\alpha})}(n^{-\beta \log n})$.*

This lower bound is based on a simple and direct reduction from solving the stochastic CSP that arises in Goldreich's proposed PRG [29] to learning DNF that is of independent interest (see Lemma 15). For comparison, the standard SQ lower bound for learning polynomial size DNF [6] relies on hardness of learning parities of size $\log n$ over the uniform distribution. Yet, parities of size $\log n$ can be easily learned from $(\log^2 n)$-wise statistical queries (since solving a system of $\log^2 n$ linear equations will uniquely identify a $\log n$-sparse parity function). Hence our lower bound holds against qualitatively stronger algorithms. Our lower bound is also exponential in the number of queries whereas the known argument implies only a quasipolynomial lower bound[1].

1.2.3 Reduction for low-communication queries

Finally, we point out that k-wise queries that require little information about each of the inputs can also be simulated using unary queries. This result is a simple corollary of the recent work of Steinhardt et al. [36] who show that any computation that extracts at most b bits from each of the samples (not necessarily at once) can be simulated using unary SQs.

▶ **Theorem 4.** *Let $\phi : X^k \to \{\pm 1\}$ be a function, and assume that ϕ has k-party public-coin randomized communication complexity of b bits per party with success probability $2/3$. Then,*

[1] We remark that an exponential lower bound on the number of queries has not been previously stated even for unary SQs. The unary version can be derived from known results as explained in Section 4.3.

there exists a randomized algorithm that, with probability at least $1-\delta$*, estimates* $\mathbb{E}_{x\sim D^k}[\phi(x)]$ *within* τ *using* $O(b\cdot k\cdot\log(1/\delta)/\tau^2)$ *queries to* $\mathsf{STAT}_D^{(1)}(\tau')$ *for some* $\tau'=\tau^{O(b)}/k$.

As a simple application of Theorem 4, we show a unary SQ algorithm that estimates the collision probability of an unknown distribution D within τ using $1/\tau^2$ queries $\mathsf{STAT}_D^{(1)}(\tau^{O(1)})$. The details appear in Section 5.

1.2.4 Corollaries for related models

Our separation result and reductions imply similar results for k-wise versions of two well-studied learning models: local differential privacy and the b-bit sampling model.

Local differentially private algorithms [30] (also referred to as randomized response) are differentially private algorithms in which each sample goes through a differentially private transformation chosen by the analyst. This model is the focus of recent privacy preserving industrial applications by Google [21] and Apple. We define a k-wise version of this model in which analyst's differentially private transformations are applied to k-tuples of samples. This model interpolates naturally between the usual (or global) differential privacy and the local model.

Kasiviswanathan et al. [30] showed that a concept class is learnable by a local differentially private algorithm if and only if it is learnable in the SQ model. Hence up to polynomial factors the models are equivalent (naturally, such polynomial factors are important for applications but here we focus only on the high-level relationships between the models). This result also implies that k-local differentially private algorithms (formally defined in Section 6.1) are equivalent to k-wise SQ algorithms (up to a polynomial blow-up in the complexity). Theorem 1 then implies an exponential separation between k-wise and $(k+1)$-wise local differentially private algorithms (see Corollary 21 for details). It can be seen as a substantial strengthening of a separation between the local model and the global one also given in [30]. The reductions in Theorem 2 and Theorem 4 imply two approaches for simulating k-local differentially private algorithms using 1-local algorithms.

The SQ model is also known to be equivalent (up to a factor polynomial in 2^b) to the b-bit sampling model introduced by Ben-David and Dichterman [4] and studied more recently in [26, 28, 43, 37, 36]. Lower bounds for the k-wise version of this model are given in [43, 36]. Our results can be easily translated to this model as well. We provide additional details in Section 6.

2 Preliminaries

For any distribution D over a domain X and any positive integer k, we denote by D^k the distribution over X^k obtained by drawing k i.i.d. samples from D. For a distribution D over a domain X and a function $\phi: X\to\mathbb{R}$, we denote $D[\phi]\doteq\mathbb{E}_{x\sim D}[\phi(x)]$.

Next, we formally define the k-wise SQ oracle.

▶ **Definition 2.1.** *Let* D *be a distribution over a domain* X *and* $\tau>0$*. A* k*-wise statistical query oracle* $\mathsf{STAT}_D^{(k)}(\tau)$ *is an oracle that given as input any function* $\phi: X^k\to[-1,+1]$*, returns some value* v *such that* $|v-\mathbb{E}_{x\sim D^k}[\phi(x)]|\le\tau$.

We say that a k-wise SQ algorithm is given access to $\mathsf{STAT}^{(k)}(\tau)$, if for every when the algorithm is given access to $\mathsf{STAT}_D^{(k)}(\tau)$, where D is the input distribution. We note that for $k=1$, Definition 2.1 reduces to the usual definition of an SQ oracle that was first introduced by Kearns [31]. The k-wise SQ complexity of solving a problem with access to $\mathsf{STAT}^{(k)}(\tau)$

is the minimum number of queries q for which exists a k-wise SQ algorithm with access to $\mathsf{STAT}^{(k)}(\tau)$ that solves the problem using at most q queries. Our discussion and results can also be easily extended to the stronger VSTAT oracle defined in [26] and to more general real-valued queries using the reductions in [24].

The PAC learning [39] is defined as follows.

▶ **Definition 2.2.** *For a class $\mathcal{C}$ of Boolean-valued functions over a domain Z, a PAC learning algorithm for $\mathcal{C}$ is an algorithm that for every P distribution over Z and $f \in \mathcal{C}$, given an error parameter $\epsilon > 0$, failure probability $\delta > 0$ and access to i.i.d. labeled examples of the form $(x, f(x))$ where $x \sim P$, outputs a hypothesis function h that, with probability at least $1 - \delta$, satisfies $\Pr_{x\sim P}[h(x) \neq f(x)] \leq \epsilon$.*

We next define one-vs-many decision problems, which will be used in the proofs in our Section 3 and Section 4.

▶ **Definition 2.3** (Decision problem $\mathcal{B}(\mathcal{D}, D_0)$)**.** *Let $\mathcal{D}$ be a set of distributions and D_0 a reference distribution over a set X. We denote by $\mathcal{B}(\mathcal{D}, D_0)$ the decision problem where we are given access to a distribution $D \in \mathcal{D} \cup \{D_0\}$ and wish to distinguish whether $D \in \mathcal{D}$ or $D = D_0$.*

3 Separation of $(k+1)$-wise from k-wise queries

We start by describing the concept class $\mathcal{C}$ that we use to prove Theorem 1. Let ℓ and k be positive integers with $\ell \geq k + 1$. The domain will be $\mathbb{F}_p^\ell$. For every $a = (a_1, \ldots, a_\ell) \in \mathbb{F}_p^\ell$, we consider the hyperplane

$$\mathsf{Hyp}_a \doteq \{z = (z_1, \ldots, z_\ell) \in \mathbb{F}_p^\ell : z_\ell = a_1 z_1 + \cdots + a_{\ell-1} z_{\ell-1} + a_\ell\}.$$

We then define the Boolean-valued function $f_a : \mathbb{F}_p^\ell \to \{\pm 1\}$ to be the indicator function of the subset $\mathsf{Hyp}_a \subseteq \mathbb{F}_p^\ell$, i.e., for every $z \in \mathbb{F}_p^\ell$,

$$f_a(z) = \begin{cases} +1 & \text{if } z \in \mathsf{Hyp}_a, \\ -1 & \text{otherwise.} \end{cases}$$

Then, we will consider the concept classes $\mathcal{C}_\ell \doteq \{f_a : a \in \mathbb{F}_p^\ell\}$. We denote $\mathcal{C} \doteq \mathcal{C}_{k+1}$. We start by stating our upper bound on the $(k+1)$-wise SQ complexity of the distribution-independent PAC learning of $\mathcal{C}_{k+1}$.

▶ **Lemma 3.1** ($(k+1)$-wise upper bound)**.** *Let p be a prime number and k be a positive integer. There exists a distribution-independent PAC learning algorithm for $\mathcal{C}_{k+1}$ that makes at most $t \cdot \log(1/\epsilon)$ queries to $\mathsf{STAT}^{(k+1)}(\epsilon/t)$, for some $t = O_k(\log p)$.*

We next state our lower bound on the k-wise SQ complexity of the same tasks considered in Lemma 3.1.

▶ **Lemma 3.2** (k-wise lower bound)**.** *Let p be a prime number and ℓ, k be positive integers with $\ell \geq k+1$ and $k = O(p)$. There exists $t = \Omega\left(p^{(\ell-k)/4}\right)$ such that any distribution-independent PAC learning alogrithm for $\mathcal{C}_\ell$ with error at most $1/2 - 2/t$ that is given access to $\mathsf{STAT}^{(k)}(1/t)$ needs at least t queries.*

Note that Lemma 3.1 and Lemma 3.2 imply Theorem 1.

3.1 Upper bound

3.1.1 Notation

We first introduce some notation that will be useful in the description of our algorithm. For any matrix M with entries in the finite field $\mathbb{F}_p$, we denote by $\mathsf{rk}(M)$ the rank of M over $\mathbb{F}_p$. Let $(a_1, \ldots, a_{k+1}) \in \mathbb{F}_p^{k+1}$ be the unknown vector that defines f_a and P be the unknown distribution over tuples $(z_1, \ldots, z_{k+1}) \in \mathbb{F}_p^{k+1}$.

Note that Hyp_a is an affine subspace of $\mathbb{F}_p^{k+1}$. To simplify our treatment of affine subspaces, we embed the points of $\mathbb{F}_p^{k+1}$ into $\mathbb{F}_p^{k+2}$ by mapping each $z \in \mathbb{F}_p^{k+1}$ to $(z, 1)$. This embedding maps every affine subspace V of $\mathbb{F}_p^{k+1}$ to a linear subspace W of $\mathbb{F}_p^{k+2}$, namely the span of the image of V under our embedding. Note that this mapping is one-to-one and allows us to easily recover V from W as $V = \{z \in \mathbb{F}_p^{k+1} \mid (z, 1) \in W\}$. Hence given $k+1$ examples

$$\big((z_{1,1}, \ldots, z_{1,k+1}), b_1\big), \big((z_{2,1}, \ldots, z_{2,k+1}), b_2\big), \ldots, \big((z_{k+1,1}, \ldots, z_{k+1,k+1}), b_{k+1}\big)$$

we define the matrix:

$$Z \doteq \begin{bmatrix} z_{1,1} & z_{1,2} & \cdot & z_{1,k+1} & 1 \\ z_{2,1} & z_{2,2} & \cdot & z_{2,k+1} & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ z_{k+1,1} & z_{k+1,2} & \cdot & z_{k+1,k+1} & 1 \end{bmatrix}. \tag{1}$$

For $\ell \in [k+1]$ we also denote by Z_ℓ the matrix that consists of the top ℓ rows of Z. Further, for a $(k+1)$-wise query function $\phi\big((z_1, b_1), \ldots, (z_{k+1}, b_{k+1})\big)$, we use Z to refer to the matrix obtained from the inputs to the function.

Let Q be the distribution defined by sampling a random example $\big((z_1, \ldots, z_{k+1}), b\big)$, conditioning on the event that $b = 1$ and outputting $(z_1, \ldots, z_{k+1}, 1)$. Note that if the examples from which Z is built are positively labeled i.i.d. examples then each row of Z is sampled i.i.d. from Q and hence Z_ℓ is distributed according to Q^ℓ. We denote by 1^{k+1} the all $+1$'s vector of length $k+1$.

3.1.2 Learning algorithm

We start by explaining the main ideas behind the algorithm. On a high level, in order to be able to use $(k+1)$-wise SQs to learn the unknown subspace, we need to make sure that there exists an affine subspace that contains most of the probability mass of the positively-labeled points and that is spanned by $k+1$ random positively-labeled points with noticeable probability. Here, the probability is with respect to the unknown distribution over labeled examples. Thus, for positively labeled tuples $(z_{1,1}, \ldots, z_{1,k+1})$, $(z_{2,1}, \ldots, z_{2,k+1})$, $\ldots$, $(z_{k+1,1}, \ldots, z_{k+1,k+1})$, we consider the $(k+1) \times (k+2)$ matrix Z defined in Equation (1). If W is the row-span of Z, then the desired (unknown) affine subspace is the set V of all points $(z_1, \ldots, z_{k+1})$ such that $(z_1, \ldots, z_{k+1}, 1) \in W$.

If the (unknown) distribution over labeled examples is such that with noticeable probability, $k+1$ random positively-labeled points form a full-rank linear system (i.e., the matrix Z has full-rank with noticeable probability conditioned on $(b_1, \ldots, b_{k+1}) = 1^{k+1}$), we can use $(k+1)$-wise SQs to find, one bit at a time, the $(k+1)$-dimensional row-span W of Z, and we can then output the set V of all points $(z_1, \ldots, z_{k+1})$ such that $(z_1, \ldots, z_{k+1}, 1) \in W$ as the desired affine subspace (below, we refer to this step as the Recovery Procedure).

We now turn to the (more challenging) case where the system is not full-rank with noticeable probability (i.e., the matrix Z is rank-deficient with high probability conditioned

Algorithm 1 $(k+1)$-wise SQ Algorithm.

Inputs. $k \in \mathbb{N}$, error probability $\epsilon > 0$.

Output. Function $f : \mathbb{F}_p^{k+1} \to \{\pm 1\}$.

Set tolerance of each SQ to $\tau = (\epsilon/2^{c\cdot(k+2)})^{(k+1)^{k+3}}$, where $c > 0$ is a large enough absolute constant.
Define the threshold $\tau_i = 2^{c\cdot(k+2-i)} \cdot k \cdot \tau^{1/(k+1)^{k+2-i}}$ for every $i \in [k+1]$.
Ask the SQ $\phi(z, b) \doteq \mathbb{1}(b = 1)$ and let w be the response.
if $w \leq \epsilon - \tau$ **then**
    Output the all -1's function.
end if
Let $\widetilde{\phi}\big((z_1, b_1), \ldots, (z_{k+1}, b_{k+1})\big) \doteq \mathbb{1}((b_1, \ldots, b_{k+1}) = 1^{k+1})$.
Ask the SQ $\widetilde{\phi}$ and let v be the response.
for $i = k+1$ down to 1 **do**
    Let $\phi_i\big((z_1, b_1), \ldots, (z_{k+1}, b_{k+1})\big) \doteq \mathbb{1}((b_1, \ldots, b_{k+1}) = 1^{k+1}$ and $\mathsf{rk}(Z) = i)$.
    Ask the SQ ϕ_i and let v_i be the response.
    if $v_i/v \geq \tau_i$ **then**
        Run Recovery Algorithm on input (i, v_i) and let $\widehat{V}$ be the subspace of $\mathbb{F}_p^{k+1}$ it outputs.
        Define function $f : \mathbb{F}_p^{k+1} \to \{-1, 1\}$ by:
        $f(z_1, \ldots, z_{k+1}) = +1$ if $(z_1, \ldots, z_{k+1}) \in \widehat{V}$.
        $f(z_1, \ldots, z_{k+1}) = -1$ otherwise.
        Return f.
    end if
end for

on $(b_1, \ldots, b_{k+1}) = 1^{k+1})$. Then, the system has rank at most i with high probability, for some $i < k+1$. There is a large number of possible i-dimensional subspaces and therefore it is no longer clear that there exists a single i-dimensional subspace that contains most of the mass of the positively-labeled points. However, we demonstrate that for every i, if the rank of Z is at most i with sufficiently high probability, then there exists a *fixed* subspace W of dimension at most i that contains a large fraction of the probability under the row-distribution of Z (it turns out that if this subspace has rank equal to i, then it should be *unique*). We can then use $(k+1)$-wise SQs to output the affine subspace V consisting of all points $(z_1, \ldots, z_{k+1})$ such that $(z_1, \ldots, z_{k+1}, 1) \in W$ (via the Recovery Procedure).

The general description of the algorithm is given in Algorithm 1, and the Recovery Procedure (allowing the reconstruction of the affine subspace V) is separately described in Algorithm 2. We denote the indicator function of event E by $\mathbb{1}(E)$. Note that the statistical query corresponding to the event $\mathbb{1}(E)$ gives an estimate of the probability of E.

3.1.3 Analysis

We now turn to the analysis of Algorithm 1 and the proof of Lemma 3.1. We will need the following lemma, which shows that if the rank of Z is at most i with high probability, then there is a *fixed* subspace of dimension at most i containing most of the probability mass under the row-distribution of Z.

▶ **Lemma 3.3.** *Let $i \in [k+1]$. If $\Pr_{Q^{k+1}}[\mathsf{rk}(Z) \leq i] \geq 1 - \xi$, then there exists a subspace W of $\mathbb{F}_p^{k+2}$ of dimension at most i such that $\Pr_{z\sim Q}[z \notin W] \leq \xi^{1/k}$.*

Algorithm 2 Recovery Procedure

Input. Integer $i \in [k+1]$.

Output. Subspace $\widehat{V}$ of $\mathbb{F}_p^{k+1}$ of dimension i.

Let $m_i = (k+2) \cdot i \cdot \lceil \log p \rceil$
for each bit $j \leq m_i$ **do**
  Define event $E_j(Z) = \mathbb{1}(\text{bit } j \text{ of row span of } Z \text{ is } 1)$.
  Let $\phi_{i,j}\big((z_1, b_1), \ldots, (z_{k+1}, b_{k+1})\big) \doteq \mathbb{1}(E_j(Z) \text{ and } (b_1, \ldots, b_{k+1}) = 1^{k+1} \text{ and } \mathsf{rk}(Z) = i)$.
  Ask the SQ $\phi_{i,j}$ and let $u_{i,j}$ be the response.
  if $u_{i,j}/v_i \geq (9/10)$ **then**
    Set bit j in binary representation of $\widehat{W}$ to 1.
  else
    Set bit j in binary representation of $\widehat{W}$ to 0.
  end if
end for
Let $\widehat{V}$ be the set all points $(z_1, \ldots, z_{k+1})$ such that $(z_1, \ldots, z_{k+1}, 1) \in \widehat{W}$.

▶ Remark. We point out that the exponential dependence on $1/k$ in the probability upper bound in Lemma 3.3 is tight. To see this, let $p = 2$, and $\{e_1, \ldots, e_k\}$ be the standard basis in $\mathbb{F}_2^k$. Consider the base distribution P on $\mathbb{F}_2^k$ that puts probability mass $1 - \alpha$ on e_1, and probability mass $\alpha/(k-1)$ on each of e_2, e_3, …, e_k. Then, a Chernoff bound implies that if we draw k i.i.d. samples from P, then the dimension of their span is at most $2 \cdot \alpha \cdot k$ with probability at least $1 - \exp(-k)$. On the other hand, for any subspace W of $\mathbb{F}_2^k$ of dimension $2 \cdot \alpha \cdot k$, the probability that a random sample from P lies inside W is only $1 - \Theta(\alpha)$.

To prove Lemma 3.3, we will use the following proposition.

▶ **Proposition 3.4.** *Let $\ell \in [k+1]$, $i \in [\ell - 1]$ and $\eta > 0$. If $\Pr_{Q^\ell}[\mathsf{rk}(Z_\ell) \leq i] \geq 1 - \eta$, then for every $\nu \in (0, 1]$, either there exists a subspace W of $\mathbb{F}_p^{k+2}$ of dimension i such that $\Pr_{z \sim Q}[z \notin W] \leq \nu$ or $\Pr_{Q^i}[\mathsf{rk}(Z_i) \leq i - 1] \geq 1 - \eta/\nu$.*

Proof. Let $p \doteq \Pr_{Q^i}[\mathsf{rk}(Z_i) \leq i - 1]$. For every (fixed) matrix $A_i \in \mathbb{F}_p^{i \times (k+2)}$, define

$$\mu(A_i) \doteq \Pr_{Q^\ell}[\mathsf{rk}(Z_\ell) \leq i \mid Z_i = A_i].$$

Then,

$$\begin{aligned}\Pr_{Q^\ell}[\mathsf{rk}(Z_\ell) \leq i] &= p + (1-p) \cdot \Pr_{Q^\ell}[\mathsf{rk}(Z_\ell) \leq i \mid \mathsf{rk}(Z_i) = i] \\ &= p + (1-p) \cdot \mathbb{E}_{Q^i}\Big[\mu(Z_i) \Big| \mathsf{rk}(Z_i) = i\Big].\end{aligned}$$

Since $\Pr_{Q^\ell}[\mathsf{rk}(Z_\ell) \leq i] \geq 1 - \eta$, we have that

$$\mathbb{E}_{Q^i}\Big[\mu(Z_i) \Big| \mathsf{rk}(Z_i) = i\Big] \geq 1 - \eta/(1-p).$$

Hence, there exists a setting $A_i \in \mathbb{F}_p^{i \times (k+2)}$ of Z_i such that $\mathsf{rk}(A_i) = i$ and

$$\Pr[\mathsf{rk}(Z_\ell) \leq i \mid Z_i = A_i] \geq 1 - \eta/(1-p).$$

We let W be the $\mathbb{F}_p$-span of the rows of A_i. Note that the dimension of W is equal to i and that $\Pr_{z \sim Q}[z \notin W] \leq \eta/(1-p)$. Thus, we conclude that for every $\nu \in (0, 1]$, either $p \geq 1 - \eta/\nu$ or $\Pr_{z \sim Q}[z \notin W] \leq \nu$, as desired. ◀

We now complete the proof of Lemma 3.3.

Proof of Lemma 3.3. Starting with $\ell = k+1$ and $\eta = \xi$, we inductively apply Proposition 3.4 with $\nu = \xi^{1/k}$ until we either get the desired subspace W or we get to the case where $i = 1$. In this case, we have that $\Pr_{Q^\ell}[\mathsf{rk}(Z_\ell) \leq 1] \geq 1 - \xi^{1/k}$ for $\ell \geq 2$. Since the last column of Z_ℓ is the all 1's vector, we conclude that there exists $z^* \in \mathbb{F}_p^{k+1}$ such that $\Pr_{z\sim Q}[z \neq (z^*, 1)] \leq \xi^{1/k}$. We can then set our subspace W to be the $\mathbb{F}_p$-span of the vector $(z^*, 1)$. ◀

For the proof of Lemma 3.1 we will also need the following lemma, which states sufficient conditions under which the Recovery Procedure (Algorithm 2) succeeds.

▶ **Lemma 3.5.** *Let $i \in [k+1]$. Assume that in Algorithm 1, $v > \epsilon^{k+1}/2$ and $v_i/v \geq \tau_i$. If there exists a subspace W of $\mathbb{F}_p^{k+2}$ of dimension equal to i such that*

$$\Pr_{z\sim Q}[z \notin W] < \frac{\tau_i}{4 \cdot (k+1)}, \tag{2}$$

then the affine subspace $\widehat{V}$ output by Algorithm 2 (i.e., the Recovery Procedure) consists of all points $(z_1, \ldots, z_{k+1})$ such that $(z_1, \ldots, z_{k+1}, 1) \in W$.

We note that Lemma 3.5 would still hold under quantitatively weaker assumptions on v, v_i/v and $\Pr_{z\sim Q}[z \notin W]$ in Equation (2). In order to keep the expressions simple, we however choose to state the above version which will be sufficient to prove Lemma 3.1. The proof of Lemma 3.5 appears in Section A.1. We are now ready to complete the proof of Lemma 3.1.

Proof of Lemma 3.1. If Algorithm 1 terminates at Step 5, then the error of the output hypothesis is at most ϵ, as desired. Henceforth, we assume that Algorithm 1 does not terminate at Step 5. Then, we have that $\Pr[b = 1] > \epsilon$, and hence $\Pr[(b_1, \ldots, b_{k+1}) = 1^{k+1}] > \epsilon^{k+1}$. Thus, the value v obtained in Step 8 of Algorithm 1 satisfies $v > \epsilon^{k+1} - \tau \geq \epsilon^{k+1}/2$, where the last inequality follows from the setting of τ. Let i^* be the first (i.e., largest) value of $i \in [k+1]$ for which $v_i/v \geq \tau_i$. To prove that such an i^* exists, we proceed by contradiction, and assume that for all $i \in [k+1]$, it is the case that $v_i/v < \tau_i$. Note that Z has an all 1's column, so it has rank at least 1. Moreover, it has rank at most $k+1$. Therefore, we have that

$$\begin{aligned}
1 &= \Pr[1 \leq \mathsf{rk}(Z) \leq k+1 \mid (b_1, \ldots, b_{k+1}) = 1^{k+1}] \\
&= \sum_{i=1}^{k+1} \Pr[\mathsf{rk}(Z) = i \mid (b_1, \ldots, b_{k+1}) = 1^{k+1}] \\
&\leq \sum_{i=1}^{k+1} \frac{v_i + \tau}{v - \tau} \\
&\leq 2 \cdot \sum_{i=1}^{k+1} \frac{v_i + \tau}{v} \\
&\leq 2 \cdot \sum_{i=1}^{k+1} \left(\frac{v_i}{v} + \frac{2\tau}{\epsilon^{k+1}}\right) \\
&< 2 \cdot \sum_{i=1}^{k+1} \tau_i + 4 \cdot (k+1) \cdot \frac{\tau}{\epsilon^{k+1}}.
\end{aligned}$$

Using the fact that τ_i is monotonically non-increasing in i and the settings of τ_1 and τ, the last inequality gives

$$1 \leq 2 \cdot (k+1) \cdot \tau_1 + 4 \cdot (k+1) \cdot \frac{\tau}{\epsilon^{k+1}} < 1,$$

a contradiction.

We now fix i^* as above. We have that

$$\begin{aligned}
\Pr[\mathsf{rk}(Z) \le i^* \mid (b_1,\dots,b_{k+1}) = 1^{k+1}] &= 1 - \sum_{i=i^*+1}^{k+1} \Pr[\mathsf{rk}(Z) = i \mid (b_1,\dots,b_{k+1}) = 1^{k+1}] \\
&\ge 1 - \sum_{i=i^*+1}^{k+1} \frac{v_i + \tau}{v - \tau} \\
&\ge 1 - 2 \cdot \sum_{i=i^*+1}^{k+1} (\frac{v_i}{v} + \frac{2\tau}{\epsilon^{k+1}}) \\
&> 1 - 2 \cdot \sum_{i=i^*+1}^{k+1} (\tau_i + 2 \cdot \frac{\tau}{\epsilon^{k+1}}) \\
&\ge 1 - 4 \cdot \sum_{i=i^*+1}^{k+1} \tau_i \\
&\ge 1 - 4 \cdot k \cdot \tau_{i^*+1}.
\end{aligned}$$

By Lemma 3.3, there exists a subspace W of $\mathbb{F}_p^{k+2}$ of dimension at most i^* such that

$$\Pr_{z \sim Q}[z \notin W] \le (4 \cdot k)^{1/k} \cdot \tau_{i^*+1}^{1/k}. \tag{3}$$

▶ **Proposition 3.6.** *For every $i \in [k]$, we have that $(k+1) \cdot (4 \cdot k)^{1/k} \cdot \tau_{i+1}^{1/k} \le \tau_i/4$.*

We note that Proposition 3.6 follows immediately from the definitions of τ_i and τ (and by letting c by a sufficiently large positive absolute constant). Moreover, Proposition 3.6 (applied with $i = i^*$) along with Equation (3) imply that $\Pr_{z \sim Q}[z \notin W]$ is at most $\tau_{i*}/(4(k+1))$.

By a union bound, we get that with probability at least

$$1 - (k+1) \cdot \Pr_{z \sim Q}[z \notin W] \ge 1 - \frac{\tau_{i^*}}{4}, \tag{4}$$

all the rows of Z belong to W.

Since $v_{i*}/v \ge \tau_{i*}$, we also have that:

$$\begin{aligned}
\Pr[\mathsf{rk}(Z) = i^* \mid (b_1,\dots,b_{k+1}) = 1^{k+1}] &\ge \frac{v_{i*} - \tau}{v + \tau} \\
&\ge \frac{1}{2} \cdot \frac{(v_{i*} - \tau)}{v} \\
&\ge \frac{1}{2} \cdot (\tau_{i^*} - \frac{2 \cdot \tau}{\epsilon^{k+1}}) \\
&\ge \frac{\tau_{i^*}}{3}
\end{aligned} \tag{5}$$

Combining Equation (4) and Equation (5), we get that the rank of W is *equal to* i^*.

Let V be the affine subspace consisting of all points $(z_1,\dots,z_{k+1})$ such that $(z_1,\dots,z_{k+1},1) \in W$. By Lemma 3.5, we get that Algorithm 2 (and hence Algorithm 1) correctly recovers the affine subspace V.

We note that the function f output by Algorithm 1 is the ± 1 indicator of a subspace of the true hyperplane Hyp_a. To see this, note that f is the ± 1 indicator function of the subspace V, and by Equations (3) and (5), we have that with probability at least $\tau_{i*}/12$ over $Z \sim Q^{k+1}$, all the columns of Z belong to W and $\mathsf{rk}(Z) = i^*$. Since the dimension of W

is equal to i^* and since we are conditioning on $(b_1, \ldots, b_{k+1}) = 1^{k+1}$, this implies that the correct label of all the points in V is $+1$. Hence, f only possibly errs on positively-labeled points (by wrongly giving them the label -1). Moreover, Algorithm 1 ensures that the output function f gives the label $+1$ to every $(z_1, \ldots, z_{k+1}) \in \mathbb{F}_p^{k+1}$ for which $(z_1, \ldots, z_{k+1}, 1) \in W$. Therefore, the function f that is output by Algorithm 1 (when it does not terminate at Step 5) has error at most the right hand side of (3). So to upper-bound the error probability, it suffices for us to verify that the right-hand side of (3) is at most ϵ. This is obtained by applying the next proposition with $i = i^* + 1$.

▶ **Proposition 3.7.** *For every $i \in [k+1]$, we have that $(4 \cdot k)^{1/k} \cdot \tau_i^{1/k} \leq \epsilon^k$.*

The proof of Proposition 3.7 follows immediately from the definitions of τ_i and τ and by letting c be a sufficiently large positive absolute constant.

The number of queries performed by the $(k+1)$-wise algorithm is at most $O(k^2 \cdot \log p)$, and their tolerance is $\tau \geq (\epsilon / 2^{c \cdot (k+2)})^{(k+1)^{k+3}}$, where c is a positive absolute constant. Finally, we remark that the dependence of the SQ complexity of the above algorithm on the error parameter ϵ is $\epsilon^{-k^{O(k)}}$. It can be improved to a linear dependence on $1/\epsilon$ by learning with error $1/3$ and then using boosting in the standard way (boosting in the SQ model works essentially as in the regular PAC model [1]). ◀

3.2 Lower bound

Our proof of lower bound is a generalization of the lower bound in [25] (for $\ell = 2$ and $k = 1$). It relies on a notion of *combined randomized statistical dimension* ("combined" refers to the fact that it examines a single parameter that lower bounds both the number of queries and the inverse of the tolerance). In order to apply this approach we need to extend it to k-wise queries. This extension follows immediately from a simple observation. If we define the domain to be $X' \doteq X^k$ and the input distribution to be $D' \doteq D^k$ then asking a k-wise query $\phi : X^k \to [-1, 1]$ to $\mathsf{STAT}_D^{(k)}(\tau)$ is equivalent to asking a unary query $\phi : X' \to [-1, 1]$ to $\mathsf{STAT}_{D'}^{(k)}(\tau)$. Using this observation we define the k-wise versions of the notions from [25] and give their properties that are needed for the proof of Lemma 3.2.

3.2.1 Preliminaries

Combined randomized statistical dimension is based on the following notion of average discrimination.

▶ **Definition 3.8** (k-wise average κ_1-discrimination)**.** *Let k be any positive integer. Let μ be a probability measure over distributions over X and D_0 be a reference distribution over X. Then,*

$$\bar{\kappa}_1^{(k)}(\mu, D_0) \doteq \sup_{\phi : X^k \to [-1,+1]} \left\{ \mathbb{E}_{D \sim \mu}[|D^k[\phi] - D_0^k[\phi]|] \right\}.$$

We denote the problem of PAC learning a concept class $\mathcal{C}$ of Boolean functions up to error ϵ by $\mathcal{L}_{PAC}(\mathcal{C}, \epsilon)$. Let Z be the domain of the Boolean functions in $\mathcal{C}$. For any distribution D_0 over labeled examples (i.e., over $Z \times \{\pm 1\}$), we define the Bayes error rate of D_0 to be

$$\mathsf{err}(D_0) = \sum_{z \in Z} \min\{D_0(z, 1), D_0(z, -1)\} = \min_{h : Z \to \{\pm 1\}} \Pr_{(z,b) \sim D_0}[h(z) \neq b].$$

▶ **Definition 3.9** (k-wise combined randomized statistical dimension)**.** *Let k be any positive integer. Let $\mathcal{D}$ be a set of distributions and D_0 a reference distribution over X. The k-wise combined randomized statistical dimension of the decision problem $\mathcal{B}(\mathcal{D}, D_0)$ is then defined as*

$$\mathsf{cRSD}^{(k)}_{\bar{\kappa}_1}(\mathcal{B}(\mathcal{D}, D_0)) \doteq \sup_{\mu \in S^{\mathcal{D}}} (\bar{\kappa}^{(k)}_1(\mu, D_0))^{-1},$$

where $S^{\mathcal{D}}$ denotes the set of all probability distributions over $\mathcal{D}$.

Further, for any concept class $\mathcal{C}$ of Boolean functions over a domain Z, and for any $\epsilon > 0$, the k-wise combined randomized statistical dimension of $\mathcal{L}_{PAC}(\mathcal{C}, \epsilon)$ is defined as

$$\mathsf{cRSD}^{(k)}_{\bar{\kappa}_1}(\mathcal{L}_{PAC}(\mathcal{C}, \epsilon)) \doteq \sup_{D_0 \in S^{Z \times \{\pm 1\}} : \mathsf{err}(D_0) > \epsilon} \mathsf{cRSD}^{(k)}_{\bar{\kappa}_1}(\mathcal{B}(\mathcal{D}_{\mathcal{C}}, D_0)),$$

where $\mathcal{D}_{\mathcal{C}} \doteq \{P^f : P \in S^Z, f \in \mathcal{C}\}$ with P^f denoting the distribution on labeled examples $(x, f(x))$ with $x \sim P$.

The next theorem lower bounds the randomized k-wise SQ complexity of PAC learning a concept class in terms of its k-wise combined randomized statistical dimension.

▶ **Theorem 5** ([25])**.** *Let $\mathcal{C}$ be a concept class of Boolean functions over a domain Z, k be a positive integer and $\epsilon, \delta > 0$. Let $d \doteq \mathsf{cRSD}^{(k)}_{\bar{\kappa}_1}(\mathcal{L}_{PAC}(\mathcal{C}, \epsilon))$. Then, the randomized k-wise SQ complexity of solving $\mathcal{L}_{PAC}(\mathcal{C}, \epsilon - 1/\sqrt{d})$ with access to $\mathsf{STAT}^{(k)}(1/\sqrt{d})$ and success probability $1 - \delta$ is at least $(1 - \delta) \cdot \sqrt{d} - 1$.*

To lower bound the statistical dimension we will use the following "average correlation" parameter introduced in [26].

▶ **Definition 3.10** (k-wise average correlation)**.** *Let k be any positive integer. Let $\mathcal{D}$ be a set of distributions and D_0 a reference distribution over X. Assume that the support of every distribution $D \in \mathcal{D}$ is a subset of the support of D_0. Then, for every $x \in X^k$, define $\hat{D}(x) \doteq \frac{D^k(x)}{D_0^k(x)} - 1$. Then, the k-wise average correlation is defined as*

$$\rho^{(k)}(\mathcal{D}, D_0) \doteq \frac{1}{|\mathcal{D}|^2} \cdot \sum_{D, D' \in \mathcal{D}} |D_0^k[\hat{D} \cdot \hat{D}']|.$$

Lemma 3.11 relates the average correlation to the average discrimination (from Definition 3.8).

▶ **Lemma 3.11** ([25])**.** *Let k be any positive integer. Let $\mathcal{D}$ be a set of distributions and D_0 a reference distribution over X. Let μ be the uniform distribution over $\mathcal{D}$. Then,*

$$\bar{\kappa}^{(k)}_1(\mu, D_0) \leq 4 \cdot \sqrt{\rho^{(k)}(\mathcal{D}, D_0)}.$$

3.2.2 Proof of Lemma 3.2

Denote $X \doteq \mathbb{F}_p^\ell \times \{\pm 1\}$. Let $\mathcal{D}$ be the set of all distributions over X^k that are obtained by sampling from any given distribution over $(\mathbb{F}_p^\ell)^k$ and labeling the k samples according to any given hyperplane indicator function f_a. Let D_0 be the uniform distribution over X^k. We now show that $\mathsf{cRSD}_{\bar{\kappa}_1}(\mathcal{B}(\mathcal{D}, D_0)) = \Omega\big(p^{(\ell-k)/2}\big)$. By definition,

$$\mathsf{cRSD}_{\bar{\kappa}_1}(\mathcal{B}(\mathcal{D}, D_0)) \doteq \sup_{\mu \in S^{\mathcal{D}}} (\bar{\kappa}_1(\mu, D_0))^{-1}.$$

We now choose the distribution μ. For $a \in \mathbb{F}_p^\ell$, we define P_a to be the distribution over $\mathbb{F}_p^\ell$ that has density $\alpha = 1/(2(p^\ell - p^{\ell-1}))$ on each of the $p^\ell - p^{\ell-1}$ points outside Hyp_a, and density $\beta = 1/p^{\ell-1} - \alpha p + \alpha = 1/(2p^{\ell-1})$ on each of the $p^{\ell-1}$ points inside Hyp_a. We then define D_a to be the distribution obtained by sampling k i.i.d. random examples of Hyp_a, the marginal of each over $\mathbb{F}_p^\ell$ being P_a. Let $\mathcal{D}' \doteq \{D_a \mid a \in \mathbb{F}_p^\ell\}$, and let μ be the uniform distribution over $\mathcal{D}'$. By Lemma 3.11, we have that $\bar{\kappa}_1(\mu, D_0) \leq 4 \cdot \sqrt{\rho(\mathcal{D}, D_0)}$, so it is enough to upper bound $\rho(\mathcal{D}, D_0)$.

We first note that for $a, a' \in \mathbb{F}_p^\ell$, we have

$$\begin{aligned}
D_0[\hat{D}_a \cdot \hat{D}_{a'}] &= \mathbb{E}_{(z,b)\sim D_0}[\hat{D}_a(z,b) \cdot \hat{D}_{a'}(z,b)] \\
&= \mathbb{E}_{(z,b)\sim D_0}\left[\left(\frac{D_a(z,b)}{D_0(z,b)} - 1\right) \cdot \left(\frac{D_{a'}(z,b)}{D_0(z,b)} - 1\right)\right] \\
&= \mathbb{E}_{(z,b)\sim D_0}\left[\frac{D_a(z,b) \cdot D_{a'}(z,b)}{D_0^2(z,b)} - \frac{D_a(z,b)}{D_0(z,b)} - \frac{D_{a'}(z,b)}{D_0(z,b)} + 1\right] \\
&= \mathbb{E}_{(z,b)\sim D_0}\left[\frac{D_a(z,b) \cdot D_{a'}(z,b)}{D_0^2(z,b)}\right] - 2 \cdot \mathbb{E}_{(z,b)\sim D_0}\left[\frac{D_a(z,b)}{D_0(z,b)}\right] + 1 \\
&= 2^{2k} \cdot p^{2k\ell} \cdot \mathbb{E}_{(z,b)\sim D_0}[D_a(z,b) \cdot D_{a'}(z,b)] \\
&\quad - 2^{k+1} \cdot p^{k\ell} \cdot \mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)] + 1
\end{aligned}$$

We now compute each of the two expectations that appear in the last equation above.

▶ **Proposition 3.12.** *For every $a \in \mathbb{F}_p^\ell$,*

$$\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)] = \frac{1}{2^k} \cdot \left(\frac{1}{p} \cdot \beta + \left(1 - \frac{1}{p}\right) \cdot \alpha\right)^k = \frac{1}{2^k \cdot p^{k\cdot\ell}}.$$

The proof of Proposition 3.12 appears in the appendix.

▶ **Proposition 3.13.** *For every $a, a' \in \mathbb{F}_p^\ell$,*

$$\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b) \cdot D_{a'}(z,b)] = \begin{cases} \frac{1}{2^k} \cdot (\frac{1}{p} \cdot \beta^2 + (1 - \frac{1}{p}) \cdot \alpha^2)^k & \text{if } \mathsf{Hyp}_a = \mathsf{Hyp}_{a'}, \\ \frac{1}{2^k} \cdot (\alpha^2 \cdot (1 - \frac{2}{p}))^k & \text{if } \mathsf{Hyp}_a \cap \mathsf{Hyp}_{a'} = \emptyset, \\ \frac{1}{2^k} \cdot (\frac{\beta^2}{p^2} + \alpha^2 \cdot (1 - \frac{2}{p} + \frac{1}{p^2}))^k & \text{otherwise.} \end{cases}$$

The proof of Proposition 3.13 appears in the appendix. Using Proposition 3.12 and Proposition 3.13, we now compute $D_0[\hat{D}_a \cdot \hat{D}_{a'}]$.

▶ **Proposition 3.14.** *For every $a, a' \in \mathbb{F}_p^\ell$,*

$$D_0[\hat{D}_a \cdot \hat{D}_{a'}] = \begin{cases} (p + 1 - \frac{1}{p-1})^k - 1 & \text{if } \mathsf{Hyp}_a = \mathsf{Hyp}_{a'}, \\ \frac{1}{2^k} \cdot \frac{(1-\frac{2}{p})^k}{(1-\frac{1}{p})^{2k}} - 1 & \text{if } \mathsf{Hyp}_a \cap \mathsf{Hyp}_{a'} = \emptyset, \\ 0 & \text{otherwise.} \end{cases}$$

The proof of Proposition 3.14 appears in the appendix. When computing $\rho(\mathcal{D}, D_0)$, we will also use the following simple proposition.

▶ **Proposition 3.15.**

1. *The number of pairs $(a, a') \in (\mathbb{F}_p^\ell)^2$ such that $\mathsf{Hyp}_a = \mathsf{Hyp}_{a'}$ is equal to p^ℓ.*
2. *The number of pairs $(a, a') \in (\mathbb{F}_p^\ell)^2$ such that Hyp_a and $\mathsf{Hyp}_{a'}$ are distinct and parallel is equal to $p^\ell \cdot (p-1)$.*

3. *The number of pairs* $(a, a') \in (\mathbb{F}_p^\ell)^2$ *such that* Hyp_a *and* $\mathsf{Hyp}_{a'}$ *are distinct and intersecting is equal to* $p^{2\cdot\ell} - p^{\ell+1}$.

Using Proposition 3.14 and Proposition 3.15, we are now ready to compute $\rho(\mathcal{D}, D_0)$ as follows

$$\begin{aligned}\rho(\mathcal{D}, D_0) &\leq \frac{1}{p^{2\cdot\ell}} \cdot \left[p^\ell \cdot (p + 1 - \frac{1}{p-1})^k + p^\ell \cdot (p-1) + p^{2\cdot\ell} \cdot 0\right] \\ &\leq O\left(\frac{1}{p^{\ell-k}}\right) + \frac{1}{p^{\ell-1}} \\ &= O\left(\frac{1}{p^{\ell-k}}\right),\end{aligned}$$

where we used above the assumption that $k = O(p)$. We deduce that $\bar{\kappa}_1(\mu, D_0) = O\left(1/p^{(\ell-k)/2}\right)$, and hence $\mathsf{cRSD}_{\bar{\kappa}_1}(\mathcal{B}(\mathcal{D}, D_0)) = \Omega\left(p^{(\ell-k)/2}\right)$. This lower bound on $\mathsf{cRSD}_{\bar{\kappa}_1}(\mathcal{B}(\mathcal{D}, D_0))$, along with Definition 3.9, Theorem 5 and the fact that D_0 has Bayes error rate equal to $1/2$, imply Lemma 3.2.

4 Reduction for flat distributions

To prove Theorem 2 we use the characterization of the SQ complexity of the problem of estimating $D^k[\phi]$ for $D \in \mathcal{D}$ using a notion of statistical dimension from [25]. Specifically, we use the characterization of the complexity of solving this problem using unary SQs and also the generalization of this characterization that characterizes the complexity of solving a problem using k-wise SQs. The latter is equal to 1 (since a single k-wise SQ suffices to estimate $D^k[\phi]$). Hence the k-wise statistical dimension is also equal to 1. We then upper bound the unary statistical dimension by the k-wise statistical dimension. The characterization then implies that an upper bound on the unary statistical dimension gives an upper bound on the SQ complexity of estimating $D^k[\phi]$.

We also give a slightly different way to define flatness that makes it easier to extend our results to other notions of divergence.

▶ **Definition 4.1.** *Let* $\mathcal{D}$ *be a set of distributions over* X. *Define*

$$R_\infty(\mathcal{D}) \doteq \inf_{\bar{D} \in S^X} \sup_{D \in \mathcal{D}} \mathrm{D}_\infty(D \| \bar{D}),$$

where S^X *denotes the set of all probability distributions over* X *and*

$$\mathrm{D}_\infty(D \| \bar{D}) \doteq \sup_{y \in X} \ln \frac{\Pr_{x \sim D}[x = y]}{\Pr_{x \sim \bar{D}}[x = y]}$$

denotes the max-divergence. We say that $\mathcal{D}$ *is* γ*-flat if* $R_\infty(\mathcal{D}) \leq \ln \gamma$.

For simplicity, we will start by relating the k-wise SQ complexity to unary SQ complexity for decision problems. The statistical dimension for this type of problems is substantially simpler than for the general problems but is sufficient to demonstrate the reduction. We then build on the results for decision problems to obtain the proof of Theorem 2.

4.1 Decision problems

The k-wise generalization of the statistical dimension for decision problems from [25] is defined as follows.

▶ **Definition 4.2.** *Let k be any positive integer. Consider a set of distributions $\mathcal{D}$ and a reference distribution D_0 over X. Let μ be a probability measure over $\mathcal{D}$ and let $\tau > 0$. The k-wise maximum covered μ-fraction is defined as*

$$\kappa_1\text{-}\mathsf{frac}^{(k)}(\mu, D_0, \tau) \doteq \sup_{\phi: X^k \to [-1,+1]} \left\{ \Pr_{D\sim\mu}[|D^k[\phi] - D_0^k[\phi]| > \tau] \right\}.$$

▶ **Definition 4.3** (k-wise randomized statistical dimension of decision problems)**.** *Let k be any positive integer. For any set of distributions $\mathcal{D}$, a reference distribution D_0 over X and $\tau > 0$, we define*

$$\mathsf{RSD}^{(k)}_{\kappa_1}(\mathcal{B}(\mathcal{D}, D_0), \tau) \doteq \sup_{\mu \in S^{\mathcal{D}}} (\kappa_1\text{-}\mathsf{frac}^{(k)}(\mu, D_0, \tau))^{-1},$$

where $S^{\mathcal{D}}$ denotes the set of all probability distributions over $\mathcal{D}$.

As shown in [25], RSD tightly characterizes the randomized statistical query complexity of solving the problem using k-wise queries. As observed before, the k-wise versions below are implied by the unary version in [25] simply by defining the domain to be $X' \doteq X^k$ and the set of input distributions to be $\mathcal{D}' \doteq \{D^k \mid D \in \mathcal{D}\}$.

▶ **Theorem 6** ([25])**.** *Let $\mathcal{B}(\mathcal{D}, D_0)$ be a decision problem, $\tau > 0, \delta \in (0, 1/2)$, $k \in \mathbb{N}$ and $d = \mathsf{RSD}^{(k)}_{\kappa_1}(\mathcal{B}(\mathcal{D}, D_0), \tau)$. Then there exists a randomized algorithm that solves $\mathcal{B}(\mathcal{D}, D_0)$ with success probability $\geq 1 - \delta$ using $d \cdot \ln(1/\delta)$ queries to $\mathsf{STAT}^{(k)}_D(\tau/2)$. Conversely, any algorithm that solves $\mathcal{B}(\mathcal{D}, D_0)$ with success probability $\geq 1 - \delta$ requires at least $d \cdot (1 - 2\delta)$ queries to $\mathsf{STAT}^{(k)}_D(\tau)$.*

We will also need the following dual formulation of the statistical dimension given in Theorem 4.3.

▶ **Lemma 4.4** ([25])**.** *Let k be any positive integer. For any set of distributions $\mathcal{D}$, a reference distribution D_0 over X and $\tau > 0$, the statistical dimension $\mathsf{RSD}^{(k)}_{\kappa_1}(\mathcal{B}(\mathcal{D}, D_0), \tau)$ is equal to the smallest d for which there exists a distribution $\mathcal{P}$ over functions from X^k to $[-1, +1]$ such that for every $D \in \mathcal{D}$,*

$$\Pr_{\phi\sim\mathcal{P}}[|D^k[\phi] - D_0^k[\phi]| > \tau] \geq \frac{1}{d}.$$

We can now state the relationship between $\mathsf{RSD}^{(k)}_{\kappa_1}$ and $\mathsf{RSD}^{(1)}_{\kappa_1}$ for any γ-flat $\mathcal{D}$.

▶ **Lemma 4.5.** *Let $\gamma \geq 1$, $\tau > 0$ and $k \in \mathbb{N}$. Let X be a domain, $\mathcal{D}$ be a γ-flat class of distributions over X and D_0 be any distribution over X. Then*

$$\mathsf{RSD}^{(1)}_{\kappa_1}(\mathcal{B}(\mathcal{D}, D_0), \tau/(2k)) \leq \frac{4k \cdot \gamma^{k-1}}{\tau} \cdot \mathsf{RSD}^{(k)}_{\kappa_1}(\mathcal{B}(\mathcal{D}, D_0), \tau).$$

Proof. Let $d \doteq \mathsf{RSD}^{(k)}_{\kappa_1}(\mathcal{B}(\mathcal{D}, D_0), \tau)$. Fact 4.4 implies the existence of a distribution $\mathcal{P}$ over k-wise functions such that for every $D \in \mathcal{D}$,

$$\Pr_{\phi\sim\mathcal{P}}[|D^k[\phi] - D_0^k[\phi]| > \tau] \geq \frac{1}{d}.$$

We now fix D and let ϕ be such that $|D^k[\phi] - D_0^k[\phi]| > \tau$.

By the standard hybrid argument,

$$\mathbb{E}_{j\sim[k]}\left[\left|D^j D_0^{k-j}[\phi] - D^{j-1} D_0^{k-j+1}[\phi]\right|\right] > \frac{\tau}{k}, \tag{6}$$

where $j \sim [k]$ denotes a random and uniform choice of j from $[k]$. This implies that

$$\mathbb{E}_{j\sim[k]}\mathbb{E}_{x_{<j}\sim D^{j-1}}\mathbb{E}_{x_{>j}\sim D_0^{k-j}}\left[\left|D[\phi(x_{<j},\cdot,x_{>j})] - D_0[\phi(x_{<j},\cdot,x_{>j})]\right|\right] > \frac{\tau}{k}.$$

By an averaging argument (and using the fact that ϕ takes values between -1 and $+1$), we get that with probability at least $\tau/(4\cdot k)$ over the choice of $j \sim [k]$, $x_{<j} \sim D^{j-1}$ and $x_{>j} \sim D_0^{k-j}$, we have that

$$\left|D[\phi(x_{<j},\cdot,x_{>j})] - D_0[\phi(x_{<j},\cdot,x_{>j})]\right| > \frac{\tau}{2\cdot k}.$$

Since $\mathcal{D}$ is a γ-flat class of distributions, there exists a (fixed) distribution $\bar{D}$ over X such that for every measurable event $E \subset X$, $\Pr_{x\sim D}[x \in E] \leq \gamma \cdot \Pr_{x\sim\bar{D}}[x \in E]$. Thus, we can replace the unknown input distribution D by the distribution $\bar{D}$ and get that, with probability at least $\tau/(4\cdot k\cdot\gamma^{k-1})$ over the choice of $j \sim [k]$, $x_{<j} \sim \bar{D}^{j-1}$ and $x_{>j} \sim D_0^{k-j}$, we have

$$\left|D[\phi(x_{<j},\cdot,x_{>j})] - D_0[\phi(x_{<j},\cdot,x_{>j})]\right| > \frac{\tau}{2\cdot k}. \tag{7}$$

We now consider the following distribution $\mathcal{P}'$ over unary SQ functions (i.e., over $[-1,+1]^X$): Independently sample ϕ from $\mathcal{P}$, j uniformly from $[k]$, $x_{<j} \sim \bar{D}^{j-1}$ and $x_{>j} \sim D_0^{k-j}$, and output the (unary) function $\phi'(x) = \phi(x_{<j}, x, x_{>j})$. Then, for every $D \in \mathcal{D}$, we have that with probability at least $\frac{1}{d}\cdot\frac{\tau}{4k}\cdot\frac{1}{\gamma^{k-1}}$ over the choice of ϕ' from $\mathcal{P}'$, we have that $|D[\phi'] - D_0[\phi']| > \tau/(2\cdot k)$. Thus, by Fact 4.4

$$\mathsf{RSD}^{(1)}_{\kappa_1}\left(\mathcal{B}(\mathcal{D},D_0),\frac{\tau}{2\cdot k}\right) \leq \frac{4d\cdot\gamma^{k-1}\cdot k}{\tau}.$$ ◀

Lemma 4.5 together with the characterization in Theorem 6 imply the following upper bound on the SQ complexity of a decision problem in terms of its k-wise SQ complexity.

▶ **Theorem 7.** *Let $\gamma \geq 1$, $\tau > 0$ and $k \in \mathbb{N}$. Let X be a domain, $\mathcal{D}$ be a γ-flat class of distributions over X and D_0 be any distribution over X. If there exists an algorithm that, with probability at least $2/3$ solves $\mathcal{B}(\mathcal{D},D_0)$ using t queries to $\mathsf{STAT}^{(k)}_D(\tau)$, then for every $\delta > 0$, there exists an algorithm that, with probability at least $1-\delta$ solves $\mathcal{B}(\mathcal{D},D_0)$ using $t\cdot 12k\cdot\gamma^{k-1}\cdot\ln(1/\delta)/\tau$ queries to $\mathsf{STAT}^{(1)}_D(\tau/(4k))$.*

4.2 General problems

We now define the general class of problems over sets of distributions and a notion of statistical dimension for these types of problems.

▶ **Definition 4.6** (Search problems). *A search problem $\mathcal{Z}$ over a class $\mathcal{D}$ of distributions and a set $\mathcal{F}$ of solutions is a mapping $\mathcal{Z}: \mathcal{D} \to 2^{\mathcal{F}} \setminus \{\emptyset\}$, where $2^{\mathcal{F}}$ denotes the set of all subsets of $\mathcal{F}$. Specifically, for every distribution $D \in \mathcal{D}$, $\mathcal{Z}(D) \subseteq \mathcal{F}$ is the (non-empty) set of valid solutions for D. For a solution $f \in \mathcal{F}$, we denote by $\mathcal{Z}_f$ the set of all distributions for which f is a valid solution.*

▶ **Definition 4.7** (Statistical dimension for search problems [25]). *For $\tau > 0$, $k \in \mathbb{N}$, a domain X and a search problem $\mathcal{Z}$ over a class of distributions $\mathcal{D}$ over X and a set of solutions $\mathcal{F}$, we define the k-*wise statistical dimension *with κ_1-discrimination τ of $\mathcal{Z}$ as*

$$\mathsf{SD}^{(k)}_{\kappa_1}(\mathcal{Z},\tau) \doteq \sup_{D_0\in S^X}\inf_{f\in\mathcal{F}}\mathsf{RSD}^{(k)}_{\kappa_1}(\mathcal{B}(\mathcal{D}\setminus\mathcal{Z}_f,D_0),\tau),$$

where S^X denotes the set of all probability distributions over X.

Lemma 8 lower-bounds the deterministic k-wise SQ complexity of a search problem in terms of its (k-wise) statistical dimension.

▶ **Theorem 8** ([25]). *Let $\mathcal{Z}$ be a search problem, $\tau > 0$ and $k \in \mathbb{N}$. The deterministic k-wise SQ complexity of solving $\mathcal{Z}$ with access to $\mathsf{STAT}^{(k)}(\tau)$ is at least $\mathsf{SD}^{(k)}_{\kappa_1}(\mathcal{Z}, \tau)$.*

The following theorem from [25] gives an upper bound on the SQ complexity of a search problem in terms of its statistical dimension. It relies on the multiplicative weights update method to reconstruct the unknown distribution sufficiently well for solving the problem. The use of this algorithm introduces dependence on KL-radius of $\mathcal{D}$. Namely, we define

$$R_{\mathrm{KL}}(\mathcal{D}) \doteq \inf_{\bar{D} \in S^X} \sup_{D \in \mathcal{D}} \mathrm{KL}(D \| \bar{D}),$$

where $\mathrm{KL}(\cdot\|\cdot)$ denotes the KL-divergence.

▶ **Theorem 9** ([25]). *Let $\mathcal{Z}$ be a search problem, $\tau, \delta > 0$ and $k \in \mathbb{N}$. There is a randomized k-wise SQ algorithm that solves $\mathcal{Z}$ with success probability $1 - \delta$ using*

$$O\left(\mathsf{SD}^{(k)}_{\kappa_1}(\mathcal{Z}, \tau) \cdot \frac{R_{\mathrm{KL}}(\mathcal{D})}{\tau^2} \cdot \log\left(\frac{R_{\mathrm{KL}}(\mathcal{D})}{\tau \cdot \delta}\right)\right)$$

queries to $\mathsf{STAT}^{(k)}(\tau/3)$.

Note that KL-divergence between two distributions is upper-bounded (and is usually much smaller) than the max-divergence we used in the definition of γ-flatness. Specifically, if $\mathcal{D}$ is γ-flat then $R_{\mathrm{KL}}(\mathcal{D}) \leq \ln \gamma$. We are now ready to prove Theorem 2 which we restate here for convenience.

▶ **Theorem 2** (restated). *Let $\gamma \geq 1$, $\tau > 0$ and k be any positive integer. Let X be a domain and $\mathcal{D}$ be a γ-flat class of distributions over X. There exists a randomized algorithm that given any $\delta > 0$ and a k-ary function $\phi : X^k \to [-1, 1]$, estimates $D^k[\phi]$ within τ for every (unknown) $D \in \mathcal{D}$ with success probability at least $1 - \delta$ using*

$$\tilde{O}\left(\frac{\gamma^{k-1} \cdot k^3}{\tau^3} \cdot \log(1/\delta)\right)$$

queries to $\mathsf{STAT}^{(1)}_D(\tau/(6 \cdot k))$.

Proof. We first observe that the task of estimating $D^k[\phi]$ up to additive τ can be viewed as a search problem $\mathcal{Z}$ over the set $\mathcal{D}$ of distributions and over the class $\mathcal{F}$ of solutions that corresponds to the interval $[-1, +1]$. Next, observe that one can easily estimate $D^k[\phi]$ up to additive τ using a single query to $\mathsf{STAT}^{(k)}_D(\tau)$. Lemma 8 implies that $\mathsf{SD}^{(k)}_{\kappa_1}(\mathcal{Z}, \tau) = 1$. By Definition 4.7, for every $D_1 \in S^X$, there exists $f \in \mathcal{F}$, such that $\mathsf{RSD}^{(k)}_{\kappa_1}(\mathcal{B}(\mathcal{D} \setminus \mathcal{Z}_f, D_1), \tau) = 1$. By Lemma 4.5,

$$\mathsf{RSD}^{(1)}_{\kappa_1}\left(\mathcal{B}(\mathcal{D} \setminus \mathcal{Z}_f, D_1), \frac{\tau}{2 \cdot k}\right) \leq \frac{4 \cdot \gamma^{k-1} \cdot k}{\tau}.$$

Thus, Fact 4.4 and Definition 4.7 imply that

$$\mathsf{SD}^{(1)}_{\kappa_1}(\mathcal{Z}, \frac{\tau}{2 \cdot k}) \leq \frac{4 \cdot \gamma^{k-1} \cdot k}{\tau}.$$

Applying Lemma 9, we conclude that there exists a randomized unary SQ algorithm that solves $\mathcal{Z}$ with probability at least $1 - \delta$ using at most

$$O\left(\gamma^{k-1} \cdot k^3 \cdot \frac{R_{\mathrm{KL}}(\mathcal{D})}{\tau^3} \cdot \log\left(\frac{k \cdot R_{\mathrm{KL}}(\mathcal{D})}{\tau \cdot \delta}\right)\right)$$

queries to $\mathsf{STAT}^{(1)}(\tau/(6\cdot k))$. This – along with the fact that $R_{\mathrm{KL}}(\mathcal{D}) \leq \ln(\gamma)$ whenever $\mathcal{D}$ is a γ-flat set of distributions – concludes the proof of Theorem 2. ◀

4.2.1 Other divergences

While the max-divergence that we used for measuring flatness suffices for the applications we give in this paper (and is relatively simple), it might be too conservative in other problems. For example, such divergence is infinite even for two Gaussian distributions with the same standard deviation but different means. A simple way to obtain a more robust version of our reduction is to use approximate max-divergence. For $\delta \in [0,1)$ it is defined as:

$$\mathrm{D}^{\delta}_{\infty}(D\|\bar{D}) \doteq \ln \sup_{E\subseteq X} \frac{\Pr_{x\sim D}[x\in E]-\delta}{\Pr_{x\sim \bar{D}}[x\in E]}.$$

Note that $\mathrm{D}^{0}_{\infty}(D\|\bar{D}) = \mathrm{D}_{\infty}(D\|\bar{D})$. Similarly, we can define a radius of $\mathcal{D}$ in this divergence

$$R^{\delta}_{\infty}(\mathcal{D}) \doteq \inf_{\bar{D}\in S^X} \sup_{D\in\mathcal{D}} \mathrm{D}^{\delta}_{\infty}(D\|\bar{D}).$$

Now, it is easy to see that, if $\mathrm{D}^{\delta}_{\infty}(D\|\bar{D}) \leq r$ then $\mathrm{D}^{k\delta}_{\infty}(D^k\|\bar{D}^k) \leq kr$. This means that if in the proof of Lemma 4.5 we use the condition $R^{\tau/(8k^2)}_{\infty}(\mathcal{D}) \leq \ln\gamma$ instead of γ-flatness then we will obtain that the event in Equation (7) holds with probability at least

$$\left(\frac{\tau}{4k} - (k-1)\cdot\frac{\tau}{8k^2}\right)/\gamma^{k-1} \geq \frac{\tau}{\gamma^{k-1}\cdot 8k}$$

over the same random choices.

This implies the following generalization of Theorem 2.

▶ **Theorem 10.** *Let $\tau > 0$ and k be any positive integer. Let $\mathcal{D}$ be a class of distributions over a domain X and $\gamma = \exp(R^{\tau/(8k^2)}_{\infty}(\mathcal{D}))$. There exists a randomized algorithm that given any $\delta > 0$ and a k-ary function $\phi : X^k \to [-1,1]$, estimates $D^k[\phi]$ within τ for every (unknown) $D \in \mathcal{D}$ with success probability at least $1-\delta$ using*

$$\tilde{O}\left(\frac{\gamma^{k-1}\cdot k^3\cdot R_{\mathrm{KL}}(\mathcal{D})}{\tau^3}\cdot\log(1/\delta)\right)$$

queries to $\mathsf{STAT}^{(1)}_{D}(\tau/(6\cdot k))$.

An alternative approach is to use Renyi divergence of order $\alpha > 1$ defined as follows:

$$\mathrm{D}_{\alpha}(D\|\bar{D}) \doteq \frac{1}{1-\alpha}\cdot\ln\left(\mathbb{E}_{y\sim D}\left[\left(\frac{\Pr_{x\sim D}[x=y]}{\Pr_{x\sim\bar{D}}[x=y]}\right)^{\alpha-1}\right]\right).$$

The corresponding radius is defined as

$$R_{\alpha}(\mathcal{D}) \doteq \inf_{\bar{D}\in S^X} \sup_{D\in\mathcal{D}} \mathrm{D}_{\alpha}(D\|\bar{D}).$$

To use it in our application we need the standard property of the Renyi divergence for product distributions $\mathrm{D}_{\alpha}(D^k\|\bar{D}^k) = k\cdot\mathrm{D}_{\alpha}(D\|\bar{D})$ and also the following simple lemma from [33, Lemma 1]:

▶ **Lemma 4.8.** *For $\alpha > 1$, any two distributions $D, \bar{D}$ over X and an event $E \subseteq X$:*

$$\Pr_{x\sim D}[x\in E] \leq \left(\exp(\mathrm{D}_{\alpha}(D\|\bar{D}))\cdot\Pr_{x\sim\bar{D}}[x\in E]\right)^{\frac{\alpha-1}{\alpha}}.$$

We will need the inverted version of this lemma:

$$\Pr_{x\sim\bar{D}}[x \in E] \geq \frac{(\Pr_{x\sim D}[x \in E])^{\frac{\alpha}{\alpha-1}}}{\exp(\mathrm{D}_\alpha(D\|\bar{D}))}.$$

Applying this in the proof of Lemma 4.5 for $\gamma = \exp(R_\alpha(\mathcal{D}))$, we obtain that the event in Equation (7) holds with probability at least

$$\left(\frac{\tau}{4k}\right)^{\frac{\alpha}{\alpha-1}} / \gamma^{k-1}.$$

This gives the following generalization of Theorem 2.

▶ **Theorem 11.** *Let $\tau > 0, \alpha > 1$ and k be any positive integer. Let $\mathcal{D}$ be a class of distributions over a domain X and $\gamma = \exp(R_\alpha(\mathcal{D}))$. There exists a randomized algorithm that given any $\delta > 0$ and a k-ary function $\phi : X^k \to [-1, 1]$, estimates $D^k[\phi]$ within τ for every (unknown) $D \in \mathcal{D}$ with success probability at least $1 - \delta$ using*

$$\tilde{O}\left(\gamma^{k-1} \cdot \left(\frac{k}{\tau}\right)^{2+\frac{\alpha}{\alpha-1}} \cdot \log(1/\delta)\right)$$

queries to $\mathsf{STAT}_D^{(1)}(\tau/(6 \cdot k))$.

4.3 Applications to solving CSPs and learning DNF

We now give some examples of the application of our reduction to obtain lower bounds against k-wise SQ algorithms. Our applications for stochastic constraint satisfaction problems (CSPs) and DNF learning. We start with the definition of a stochastic CSP with a *planted solution* which is a pseudo-random generator based on Goldreich's proposed one-way function [29].

▶ **Definition 12.** Let $t \in \mathbb{N}$ and $P : \{\pm 1\}^t \to \{\pm 1\}$ be a fixed predicate. We are given access to samples from a distribution P_σ, corresponding to a ("planted") assignment $\sigma \in \{\pm 1\}^n$. A sample from this distribution is a uniform-random t-tuple $(i_1, \ldots, i_t)$ of distinct variable indices along with the value $P(\sigma_{i_1}, \ldots, \sigma_{i_t})$. The goal is to recover the assignment σ when given m independent samples from P_σ. A (potentially) easier problem is to distinguish any such planted distribution from the distribution U_t in which the value is an independent uniform-random coin flip (instead of $P(\sigma_{i_1}, \ldots, \sigma_{i_t})$).

We say that a predicate $P : \{\pm 1\}^t \to \{\pm 1\}$ has complexity r if r is the degree of the lowest-degree non-zero Fourier coefficient of P. It can be as large as t (for the parity function). A lower bound on the (unary) SQ complexity of solving such CSPs was shown by [28] (their result is for the stronger VSTAT oracle but here we state the version for the STAT oracle).

▶ **Theorem 13** ([28]). *Let $t, q \in \mathbb{N}$ and $P : \{\pm 1\}^t \to \{\pm 1\}$ be a fixed predicate of complexity r. Then for any $q > 0$, any algorithm that, given access to a distribution $D \in \{P_\sigma \mid \sigma \in \{\pm 1\}^n\} \cup \{U_t\}$ decides correctly whether $D = P_\sigma$ or $D = U_t$ with probability at least $2/3$ needs $q/2^{O(t)}$ queries to $\mathsf{STAT}_D^{(1)}\left(\left(\frac{\log q}{n}\right)^{r/2}\right)$.*

The set of input distributions in this problem is 2-flat relative to U_t and it is one-to-many decision problem. Hence Theorem 7 implies[2] the following lower bound for k-wise SQ algorithms.

[2] We can also get essentially the same result by applying the simulation of a k-wise SQ using unary SQs from Theorem 2.

▶ **Theorem 14.** *Let $t \in \mathbb{N}$ and $P : \{\pm 1\}^t \to \{\pm 1\}$ be a fixed predicate of complexity r. Then for any $\alpha > 0$, any algorithm that, given access to a distribution $D \in \{P_\sigma \mid \sigma \in \{\pm 1\}^n\} \cup \{U_t\}$ decides correctly whether $D = P_\sigma$ or $D = U_t$ with probability at least $2/3$ needs $2^{n^{1-\alpha}-O(t)}$ queries to $\mathsf{STAT}_D^{(n^{1-\alpha})}\left((2/n^\alpha)^{r/2} \cdot n^{1-\alpha}/4\right)$.*

Proof. Let $\mathcal{A}$ be a k-wise SQ algorithm using q' queries to $\mathsf{STAT}_D^{(n^{1-\alpha})}\left((2/n^\alpha)^{r/2} \cdot n^{1-\alpha}/6\right)$ which solves the problem with success probability $2/3$. We let $k = n^{1-\alpha}$ and apply Theorem 7 to obtain an algorithm that uses unary SQs and solves the problem with success probability $2/3$. This algorithm uses $q_0 = q' \cdot 2^{n^{1-\alpha}} \cdot n^{O(r)}$ queries to $\mathsf{STAT}_D^{(1)}\left((2/n^\alpha)^{r/2}\right)$. Now choosing $q = 2^{2n^{1-\alpha}}$ we get that $\left(\frac{\log q}{n}\right)^{r/2} \leq (2/n^\alpha)^{r/2}$. This means that $q_0 \geq q/2^{O(t)} = 2^{2n^{1-\alpha}-O(t)}$. Hence $q' = 2^{2n^{1-\alpha}-O(t)-n^{1-\alpha}-O(r)} = 2^{n^{1-\alpha}-O(t)}$. ◀

Similar lower bounds can be obtained for other problems considered in [28], namely, planted satisfiability and t-SAT refutation.

To obtain a lower bound for learning DNF formulas we can use a simple reduction from the Goldreich's PRG defined above to learning DNF formulas of polynomial size. It is based on ideas implicit in the reduction from t-SAT refutation to DNF learning from [13].

▶ **Lemma 15.** *$P : \{\pm 1\}^t \to \{\pm 1\}$ be a fixed predicate. There exists a mapping M from t-tuples of indices in $[n]$ to $\{0,1\}^{tn}$ such that for every $\sigma \in \{\pm 1\}^n$ there exists a DNF formula f_σ of size 2^t satisfying $P(\sigma_{i_1}, \ldots, \sigma_{i_t}) = f_\sigma(M(i_1, \ldots, i_t))$.*

Proof. The mapping M maps $(i_1, \ldots, i_t)$ to the concatenation of the indicator vectors of each of the indices. Namely, for $j \in [t]$ and $\ell \in [n]$, $M(i_1, \ldots, i_t)_{j,\ell} = 1$ if and only if $i_j = \ell$, where we use the double index j, ℓ to refer to element $n(j-1)+\ell$ of the vector. Let $v_{j,\ell}$ denote the variable with the index j, ℓ. Let σ be any assignment and we denote by z_j^σ the j-th variable of our predicate P when the assignment is equal to σ. We first observe that $z_j^\sigma \equiv \bigwedge_{\ell \in [n], \sigma_\ell = 0} \bar{v}_{j,\ell}$. This is true since, by definition, the value of the j-th variable of our predicate is σ_{i_j}. This value is 1 if and only if $i_j \notin \{\ell \in [n] \mid \sigma_\ell = 0\}$. This is equivalent to $v_{j,\ell}$ being equal to 0 for all $\ell \in [n]$ such that $\sigma_\ell = 0$. Analogously, $\bar{z}_j^\sigma \equiv \bigwedge_{\ell \in [n], \sigma_\ell = 1} \bar{v}_{j,\ell}$. This implies that any conjunction of variables $z_1^\sigma, \bar{z}_1^\sigma, \ldots, z_t^\sigma, \bar{z}_t^\sigma$ can be expressed as a conjunction over variables $\bar{v}_{j,\ell}$. Any predicate P can be expressed as a disjunction of at most 2^t conjunctions and hence there exists a DNF formula f_σ of size at most 2^t whose value on $M(i_1, \ldots, i_t)$ is equal to $P(\sigma_{i_1}, \ldots, \sigma_{i_t})$. ◀

This reduction implies that by converting a sample $((i_1, \ldots, i_t), b)$ to a sample $(M(i_1, \ldots, i_t), b)$ we can transform the Goldreich's PRG problem into a problem in which our goal is to distinguish examples of some DNF formula f_σ from randomly labeled examples. Naturally, an algorithm that can learn DNF formulas can output a hypothesis which predicts the label (with some non-trivial accuracy), whereas such hypothesis cannot exist for predicting random labels. Hence known SQ lower bounds on planted CSPs [28] immediately imply lower bounds for learning DNF. Further, by applying Lemma 15 together with Thm. 14 for $t = r = \log n$ we obtain the first lower bounds for learning DNF against $n^{1-\alpha}$-wise SQ algorithms.

▶ **Theorem 16.** *For any constant (independent of n) $\alpha > 0$, there exists a constant $\beta > 0$ such that any algorithm that PAC learns DNF formulas of size n with error $< 1/2 - n^{-\beta \log n}$ and success probability at least $2/3$ needs at least $2^{n^{1-\alpha}}$ queries to $\mathsf{STAT}_D^{(n^{1-\alpha})}(n^{-\beta \log n})$.*

We remark that this is a lower bound for PAC learning polynomial size DNF formulas with respect to some fixed (albeit non-uniform) distribution over $\{0,1\}^n$. The approach for relating k-wise SQ complexity to unary SQ complexity given in [9] applies to this setting. Yet, in their proof the tolerance needed for the unary SQ algorithm is $\tau/2^k$ and therefore it would not give a non-trivial lower bounds beyond $k = O(\log n)$.

5 Reduction for low-communication queries

In this section, we prove Theorem 4 using a recent result of Steinhardt, Valiant and Wager [36]. Their result can be seen giving a SQ algorithm that simulates a communication protocol between n parties. Each party is holding a sample drawn i.i.d. from distribution D and broadcasts at most b bits about its sample (to all the other parties). The bits can be sent over multiple rounds. This is essentially the standard model of multi-party communication complexity (*e.g.* [32]) but with the goal of solving some problem about the unknown distribution D rather than computing a specific function of the inputs. Alternatively, one can also see this model as a single algorithm that extracts at most b-bits of information about each random sample from D and is allowed to extract the bits in an arbitrary order (generalizing the b-bit sampling model that we discuss in Section 6.2 and in which b-bits are extracted from each sample at once). We refer to this model simply as algorithms that extract at most b bits per sample.

▶ **Theorem 17** ([36]). *Let $\mathcal{A}$ be an algorithm that uses n samples drawn i.i.d. from a distribution D and extracts at most b bits per sample. Then, for every $\beta > 0$, there is an algorithm $\mathcal{B}$ that makes at most $2 \cdot b \cdot n$ queries to $\mathsf{STAT}_D^{(1)}(\beta/(2^{b+1} \cdot k))$ and the output distributions of $\mathcal{A}$ and $\mathcal{B}$ are within total variation distance β.*

We will use this simulation to estimate the expectation of k-wise functions that have low communication complexity. Specifically, we recall the following standard model of public-coin randomized k-party communication complexity.

▶ **Definition 5.1.** *For a function $\phi : X^k \to \{\pm 1\}$ we say that ϕ has a k-party public-coin randomized communication complexity of at most b bits per party with success probability $1 - \delta$ if there exist a protocol satisfying the following conditions. Each of the parties is given $x_i \in X$ and access to shared random bits. In each round one of the parties can compute one or more bits using its input, random bits and all the previous communication and then broadcast it to all the other parties. In the last round one of the parties computes a bit that is the output of the protocol. Each of the parties communicates at most b bits in total. For every $x_1, \ldots, x_k \in X$, with probability at least $1 - \delta$ over the choice of the random bits the output of the protocol is equal to $\phi(x_1, \ldots, x_k)$.*

We are now ready to prove Theorem 4 which we restate here for convenience.

▶ **Theorem 4** (restated). *Let $\phi : X^k \to \{\pm 1\}$ be a function, and assume that ϕ has k-party public-coin randomized communication complexity of b bits per party with success probability $2/3$. Then, there exists a randomized algorithm that, with probability at least $1 - \delta$, estimates $\mathbb{E}_{x \sim D^k}[\phi(x)]$ within τ using $O(b \cdot k \cdot \log(1/\delta)/\tau^2)$ queries to $\mathsf{STAT}_D^{(1)}(\tau')$ for some $\tau' = \tau^{O(b)}/k$.*

Proof. We first amplify the success probability of the protocol for computing ϕ to $\delta' \doteq \tau/8$ using the majority vote of $O(\log(1/\delta'))$ repetitions. By Yao's minimax theorem [42] there exists a deterministic protocol Π' that succeeds with probability at least $1 - \delta'$ for $(x_1, \ldots, x_k) \sim D^k$. Applying Theorem 17, we obtain a unary SQ algorithm $\mathcal{A}$ whose output

is within total variation distance at most $\beta \doteq \tau/8$ from $\Pi'(x_1, \ldots, x_k)$ (and we can assume that the output of $\mathcal{A}$ is in $\{\pm 1\}$). Therefore:

$$|\mathbb{E}[\mathcal{A}] - D^k[\phi]| \leq |\mathbb{E}[\mathcal{A}] - \mathbb{E}_{D^k}[\Pi'(x_1, \ldots, x_k)]| + |\mathbb{E}_{D^k}[\Pi'(x_1, \ldots, x_k)] - D^k[\phi]| \leq \frac{2\tau}{8} + \frac{2\tau}{8} = \frac{\tau}{2}.$$

Repeating $\mathcal{A}$ $O(\log(1/\delta)/\tau^2)$ times and taking the mean, we get an estimate of $D^k[\phi]$ within τ with probability at least $1-\delta$. This algorithm uses $O(b \cdot k \cdot \log(1/\delta)/\tau^2)$ queries to $\mathsf{STAT}_D^{(1)}(\tau')$ for $\tau' = \frac{\tau}{8}/(2^{O(\log(8/\tau)\cdot b)} \cdot k) = \tau^{O(b)}/k$. ◀

The collision probability for a distribution D is defined as $\Pr_{(x_1,x_2)\sim D^2}[x_1 = x_2]$. This corresponds to $\phi(x_1, x_2)$ being the Equality function which, as is well-known, has randomized 2-party communication complexity of $O(1)$ bits per party with success probability $2/3$ (see, e.g., [32]). Applying Theorem 4 with $k = 2$ we get the following corollary.

▶ **Corollary 18.** *For any $\tau, \delta > 0$, there is a SQ algorithm that estimates the collision probability of an unknown distribution D within τ with success probability $1 - \delta$ using $O(\log(1/\delta)/\tau^2)$ queries to $\mathsf{STAT}_D^{(1)}(\tau^{O(1)})$.*

6 Corollaries for other models

6.1 k-local differential privacy

We start by formally defining the k-wise version of the *local differentially privacy* model from [30].

▶ **Definition 6.1** (k-local randomizer)**.** *A k-local ϵ-differentially private (DP) randomizer is a randomized map $R : X^k \to W$ such that for all $u, u' \in X^k$ and all $w \in W$, we have that $\Pr[R(u) = w] \leq e^\epsilon \cdot \Pr[R(u') = w]$ where the probabilities are taken over the coins of R.*

The following definition gives a k-wise generalization of the local randomizer (LR) oracle which was used in [30].

▶ **Definition 6.2** (k-local Randomizer Oracle)**.** *Let $z = (z_1, \ldots, z_n) \in X^n$ be a database. A k-LR oracle $\mathsf{LR}_z(\cdot, \cdot)$ gets a k-tuple of indices $\bar{i} \in [n]^k$ and a k-local ϵ-DP randomizer as inputs, and outputs an element $w \in W$ which is sampled from the distribution $R(z_{i_1}, \ldots, z_{i_k})$.*

We are now ready to give the definition of k-local differential privacy.

▶ **Definition 6.3** (k-local differentially private algorithm)**.** *A k-local ϵ-differentially private algorithm is an algorithm that accesses a database $z \in X^n$ via a k-LR oracle LR_z with the restriction that for all $i \in [n]$, if $\mathsf{LR}_z(\bar{i}_1, R_1), \ldots, \mathsf{LR}_z(\bar{i}_t, R_t)$ are the algorithm's invocations of LR_z on k-tuples of indices that include index i, where for each $j \in [t]$ R_j is a k-local ϵ_j-DP randomizer, then $\epsilon_1 + \cdots + \epsilon_t \leq \epsilon$.*

The following two theorems – which follow from Theorem 5.7 and Lemma 5.8 of [30] – show that k-local differentially private algorithms are equivalent (up to polynomial factors) to k-wise statistical query algorithms.

▶ **Theorem 19.** *Let $\mathcal{A}_{SQ}$ be a k-wise SQ algorithm that makes at most t queries to $\mathsf{STAT}_D^{(k)}(\tau)$. Then, for every $\beta > 0$, there exists a k-local ϵ-DP algorithm $\mathcal{A}_{DP}$ such that if the database z has $n \geq n_0 = O(k \cdot t \cdot \log(t/\beta)/(\epsilon^2 \cdot \tau^2))$ entries sampled i.i.d. from the distribution D, then $\mathcal{A}_{DP}$ makes n_0/k queries and the total variation between $\mathcal{A}_{DP}$'s and $\mathcal{A}_{SQ}$'s output distributions is at most β.*

▶ **Theorem 20.** *Let $z \in X^n$ be a database with entries drawn i.i.d. from a distribution D. For every k-local ϵ-DP algorithm $\mathcal{A}_{DP}$ making t queries to LR_z and $\beta > 0$, there exists a k-wise statistical query algorithm $\mathcal{A}_{SQ}$ that in expectation makes $O(t \cdot e^{\epsilon})$ queries to $\mathsf{STAT}_D^{(k)}(\tau)$ for $\tau = \Theta(\beta/(e^{2\epsilon} \cdot t))$ such that the total variation between $\mathcal{A}_{SQ}$'s and $\mathcal{A}_{DP}$'s output distributions is at most β.*

By combining Theorem 1, Theorem 19 and Theorem 20 we then obtain the following corollary.

▶ **Corollary 21.** *For every positive integer k and any prime number p, there is a concept class $\mathcal{C}$ of Boolean functions defined over a domain of size p^{k+1} for which there exists a $(k+1)$-local 1-DP distribution-independent PAC learning algorithm using a database consisting of $\widetilde{O}_k(\log p)$ i.i.d. samples, whereas any k-local 1-DP distribution-independent PAC learning algorithm requires $\Omega_k(p^{1/4})$ samples.*

The reduction in Theorem 2 then implies that for γ-flat classes of distributions a k-local DP algorithm can be simulated by a 1-local DP algorithm with an overhead that is linear in γ^{k-1} and polynomial in other parameters.

▶ **Theorem 22.** *Let $\gamma \geq 1$, k be any positive integer. Let X be a domain and $\mathcal{D}$ a γ-flat class of distributions over X. Let $z \in X^n$ be a database with entries drawn i.i.d. from a distribution $D \in \mathcal{D}$. For every k-local ϵ-DP algorithm $\mathcal{A}$ making t queries to a k-LR oracle LR_z and $\beta > 0$, there exists a 1-local ϵ-DP algorithm $\mathcal{B}$ such that if $n \geq n_0 = \tilde{O}\left(\frac{\gamma^{k-1} \cdot t^6 \cdot k^6 \cdot e^{11\epsilon}}{\beta^3 \epsilon^2}\right)$ then for every $D \in \mathcal{D}$, $\mathcal{B}$ makes n_0/k queries to 1-LR oracle LR'_z and the total variation distance between $\mathcal{B}$'s and $\mathcal{A}$'s output distributions is at most β.*

The reduction from Theorem 4 can be translated to this model analogously.

6.2 k-wise b-bit sampling model

For an integer $b > 0$, a b-bit sampling oracle $\mathrm{BS}_D(b)$ is defined as follows: Given any function $\phi : X \to \{0,1\}^b$, $\mathrm{BS}_D(b)$ returns $\phi(x)$ for x drawn randomly and independently from D, where D is the unknown input distribution. This oracle was first studied by Ben-David and Dichterman [4] as a *weak Restricted Focus of Attention* model. They showed that algorithms in this model can be simulated efficiently using statistical queries and vice versa. Lower bounds against algorithms that use such an oracle have been studied in [26, 28]. More recently, motivated by communication constraints in distributed systems, the sample complexity of several basic problems in statistical estimation has been studied in this and related models [43, 37, 36]. These works also study the natural k-wise generalization of this model. Specifically, $\mathrm{BS}_D^{(k)}(b)$ is the oracle that given any function $\phi : X^k \to \{0,1\}^b$, returns $\phi(x)$ for x drawn randomly and independently from D^k.

The following two theorems – which follow from Theorem 5.2 in [4] and Proposition 3 in [36] (that strengthens a similar result in [4]) – show that k-wise algorithms in the b-bit sampling model are equivalent (up to polynomial and 2^b factors) to k-wise statistical query algorithms.

▶ **Theorem 23.** *Let $\mathcal{A}_{SQ}$ be a k-wise SQ algorithm that makes at most t Boolean queries to $\mathsf{STAT}_D^{(k)}(\tau)$. Then, for every $\beta > 0$, there exists a k-wise 1-bit sampling algorithm $\mathcal{A}_{1\text{-}bit}$ that uses $O(\frac{t}{\tau^2} \cdot \log(t/\beta))$ queries to $BS_D^{(k)}(b)$ and the total variation distance between $\mathcal{A}_{SQ}$'s and $\mathcal{A}_{1\text{-}bit}$'s output distributions is at most β.*

▶ **Theorem 24.** *Let $\mathcal{A}_{b\text{-}bit}$ be a k-wise b-bit sampling algorithm that makes at most t queries to $BS_D^{(k)}(b)$. Then, for every $\beta > 0$, there exists a k-wise SQ algorithm $\mathcal{A}_{SQ}$ that makes $2bt$ queries to $\mathsf{STAT}_D^{(k)}(\beta/(2^{b+1}t))$ and the total variation distance between $\mathcal{A}_{SQ}$'s and $\mathcal{A}_{b\text{-}bit}$'s output distributions is at most β.*

Feldman et al. [26] give a tighter correspondence between the BS oracle and the slightly stronger VSTAT oracle. Their simulations can be extended to the k-wise case in a similar way.

The following corollary now follows by combining Theorem 1, Theorem 23 and Theorem 24.

▶ **Corollary 25.** *Let $b = O(1)$. For every positive integer k and any prime number p, there is a concept class $\mathcal{C}$ of Boolean functions defined over a domain of size p^{k+1} for which there exists a $(k+1)$-wise b-bit sampling distribution-independent PAC learning algorithm making $\widetilde{O}_k(\log p)$ queries, whereas any k-wise b-bit sampling distribution-independent PAC learning algorithm requires $\widetilde{\Omega}_k(p^{1/12})$ queries.*

The reduction in Theorem 2 then implies that for γ-flat classes of distributions a k-wise 1-bit sampling algorithm can be simulated by a 1-wise 1-bit sampling algorithm.

▶ **Theorem 26.** *Let $\gamma \geq 1$, k be any positive integer. Let X be a domain and $\mathcal{D}$ a γ-flat class of distributions over X. For every algorithm $\mathcal{A}$ making t queries to $BS_D^{(k)}(1)$ and every $\beta > 0$, there exists a 1-bit sampling algorithm $\mathcal{B}$ that for every $D \in \mathcal{D}$, uses $\tilde{O}\left(\frac{\gamma^{k-1} \cdot t^6 \cdot k^5}{\beta^3}\right)$ queries to $BS_D(1)$ and the total variation distance between $\mathcal{B}$'s and $\mathcal{A}$'s output distributions is at most β.*

References

1. Javed A Aslam and Scott E Decatur. General bounds on statistical query learning and pac learning with noise via hypothesis boosting. In *Foundations of Computer Science, 1993. Proceedings., 34th Annual Symposium on*, pages 282–291. IEEE, 1993.
2. Maria-Florina Balcan, Avrim Blum, Shai Fine, and Yishay Mansour. Distributed learning, communication complexity and privacy. In Shie Mannor, Nathan Srebro, and Robert C. Williamson, editors, *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, volume 23 of *JMLR Proceedings*, pages 26.1–26.22. JMLR.org, 2012. URL: `http://www.jmlr.org/proceedings/papers/v23/balcan12a/balcan12a.pdf`.
3. Maria-Florina Balcan and Vitaly Feldman. Statistical active learning algorithms for noise tolerance and differential privacy. *Algorithmica*, 72(1):282–315, 2015. `doi:10.1007/s00453-014-9954-9`.
4. Shai Ben-David and Eli Dichterman. Learning with restricted focus of attention. *J. Comput. Syst. Sci.*, 56(3):277–298, 1998. `doi:10.1006/jcss.1998.1569`.
5. Shai Ben-David, Alon Itai, and Eyal Kushilevitz. Learning by distances. In Mark A. Fulk and John Case, editors, *Proceedings of the Third Annual Workshop on Computational Learning Theory, COLT 1990, University of Rochester, Rochester, NY, USA, August 6-8, 1990.*, pages 232–245. Morgan Kaufmann, 1990. URL: `http://dl.acm.org/citation.cfm?id=92644`.
6. A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of STOC*, pages 253–262, 1994.

7 Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In Chen Li, editor, *Proceedings of the Twenty-fourth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 13-15, 2005, Baltimore, Maryland, USA*, pages 128–138. ACM, 2005. doi:10.1145/1065167.1065184.

8 Avrim Blum, Alan Frieze, Ravi Kannan, and Santosh Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1-2):35–52, 1998.

9 Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003.

10 D. Boneh and R. Lipton. Amplification of weak learning over the uniform distribution. In *Proceedings of the Sixth Annual Workshop on Computational Learning Theory*, pages 347–351, 1993.

11 Cheng Chu, Sang Kyun Kim, Yi-An Lin, YuanYuan Yu, Gary Bradski, Andrew Y Ng, and Kunle Olukotun. Map-reduce for machine learning on multicore. *Advances in neural information processing systems*, 19:281, 2007.

12 Dana Dachman-Soled, Vitaly Feldman, Li-Yang Tan, Andrew Wan, and Karl Wimmer. Approximate resilience, monotonicity, and the complexity of agnostic learning. In *Proceedings of SODA*, 2015.

13 Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limitations on learning dnf's. In *COLT*, pages 815–830, 2016. URL: http://jmlr.org/proceedings/papers/v49/daniely16.html.

14 Anindya De, Ilias Diakonikolas, and Rocco A Servedio. Learning from satisfying assignments. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 478–497. SIAM, 2015.

15 Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. *CoRR*, abs/1611.03473, 2016. URL: http://arxiv.org/abs/1611.03473.

16 Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In Frank Neven, Catriel Beeri, and Tova Milo, editors, *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 9-12, 2003, San Diego, CA, USA*, pages 202–210. ACM, 2003. doi:10.1145/773153.773173.

17 John Dunagan and Santosh Vempala. A simple polynomial-time rescaling algorithm for solving linear programs. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 315–320. ACM, 2004. doi:10.1145/1007352.1007404.

18 Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toni Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. In *Advances in Neural Information Processing Systems*, pages 2341–2349, 2015.

19 Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 117–126. ACM, 2015.

20 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006. doi:10.1007/11681878_14.

21 Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. RAPPOR: randomized aggregatable privacy-preserving ordinal response. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 1054–1067, 2014.

22 V. Feldman, H. Lee, and R. Servedio. Lower bounds and hardness amplification for learning shallow monotone formulas. In *COLT*, volume 19, pages 273–292, 2011.

23 Vitaly Feldman. Evolvability from learning algorithms. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 619–628. ACM, 2008.

24 Vitaly Feldman. Dealing with range anxiety in mean estimation via statistical queries. *arXiv*, abs/1611.06475, 2016. URL: `http://arxiv.org/abs/1611.06475`.

25 Vitaly Feldman. A general characterization of the statistical query complexity. *CoRR*, abs/1608.02198, 2016. URL: `http://arxiv.org/abs/1608.02198`.

26 Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. *arXiv, CoRR*, abs/1201.1214, 2012. Extended abstract in STOC 2013.

27 Vitaly Feldman, Cristobal Guzman, and Santosh Vempala. Statistical query algorithms for mean vector estimation and stochastic convex optimization. *CoRR*, abs/1512.09170, 2015. Extended abstract in SODA 2017. URL: `http://arxiv.org/abs/1512.09170`.

28 Vitaly Feldman, Will Perkins, and Santosh Vempala. On the complexity of random satisfiability problems with planted solutions. *CoRR*, abs/1311.4821, 2013. Extended abstract in STOC 2015.

29 Oded Goldreich. Candidate one-way functions based on expander graphs. *IACR Cryptology ePrint Archive*, 2000:63, 2000.

30 Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

31 Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

32 Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

33 Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Multiple source adaptation and the rényi divergence. In *UAI*, pages 367–374, 2009. URL: `https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1600&proceeding_id=25`.

34 Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 765–774. ACM, 2010.

35 Indrajit Roy, Srinath TV Setty, Ann Kilzer, Vitaly Shmatikov, and Emmett Witchel. Airavat: Security and privacy for mapreduce. In *NSDI*, volume 10, pages 297–312, 2010.

36 J. Steinhardt, G. Valiant, and S. Wager. Memory, communication, and statistical queries. In *COLT*, pages 1490–1516, 2016.

37 Jacob Steinhardt and John C. Duchi. Minimax rates for memory-bounded sparse linear regression. In *COLT*, pages 1564–1587, 2015. URL: `http://jmlr.org/proceedings/papers/v40/Steinhardt15.html`.

38 Arvind Sujeeth, HyoukJoong Lee, Kevin Brown, Tiark Rompf, Hassan Chafi, Michael Wu, Anand Atreya, Martin Odersky, and Kunle Olukotun. Optiml: an implicitly parallel domain-specific language for machine learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 609–616, 2011.

39 Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

40 Leslie G Valiant. Evolvability. *Journal of the ACM (JACM)*, 56(1):3, 2009.

41 Larry Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

42 Andrew Yao. Probabilistic computations: Toward a unified measure of complexity. In *FOCS*, pages 222–227, 1977.

43 Yuchen Zhang, John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Information-theoretic lower bounds for distributed statistical estimation with communication constraints. In *Proceedings of NIPS*, pages 2328–2336, 2013.

A Omitted proofs

A.1 Proof of Lemma 3.5

In the following, we denote by $o_c(\cdot)$ and $\omega_c(\cdot)$ asymptotic functions obtained by taking the limit as the parameter c goes to infinity. In particular, $o_c(1)$ can be made arbitrarily close to 0 by letting c be large enough.

Let W be as in the statement of Lemma 3.5. To prove the lemma, it suffices to show that each bit j in the binary representation of the subspace $\widehat{W}$ constructed by Algorithm 2 is equal to the corresponding bit of W. Henceforth, we fix j. We consider the two cases where bit j of W is equal to 1, and where it is equal to 0.

First, we assume that bit j of W is equal to 1, and prove that in the execution of Algorithm 2, it will be the case that $u_{i,j}/v_i \geq 1 - o_c(1)$. We can then set c to be sufficiently large to ensure that $u_{i,j}/v_i \geq (9/10)$. Note that for any positive real numbers N, D and τ such that $\tau = o(N)$ and $\tau = o(D)$, we have that

$$\frac{N-\tau}{D+\tau} \geq \frac{N}{D} \cdot (1 - o(1)).$$

Thus, it is enough to show that the next three statements hold:

(i) $\tau = o_c(\overline{v}_i)$,
(ii) if bit j of W is 1, then $(\overline{u}_{i,j}/\overline{v}_i) \geq 1 - o_c(1)$,
(iii) if bit j of W is 1, then $\tau = o_c(\overline{u}_{i,j})$,

where $\overline{u}_{i,j} \triangleq \mathbb{E}[\phi_{i,j}]$ and $\overline{v}_i \triangleq \mathbb{E}[\phi_i]$.

To show (i) above, note that

$$\begin{aligned}
\overline{v}_i &= \Pr\left[(b_1, \ldots, b_{k+1}) = 1^{k+1} \text{ and } \mathsf{rk}(Z) = i\right] \\
&\geq v_i - \tau \\
&\geq v \cdot \tau_i - \tau \\
&\geq \omega_c(\tau),
\end{aligned}$$

where the first inequality follows from the definition of v_i and the SQ guarantee, the second inequality follows from the given assumption (in the statement of Lemma 3.5) that $(v_i/v) \geq \tau_i$, and the last inequality follows from the fact that since $v > \epsilon^{k+1}/2$, for every $i \in [k+1]$, we have that

$$\tau = o_c\Big((v \cdot \tau_i - \tau) \cdot (1 - \tau_i/4)\Big).$$

.

Recall the definition of the event $E_j(Z)$ from the description of Algorithm 2. To show (ii) above, note that

$$\begin{aligned}
\frac{\overline{u}_{i,j}}{\overline{v}_i} &= \Pr\left[E_j(Z) \mid (b_1,\ldots,b_{k+1}) = 1^{k+1} \text{ and } \mathsf{rk}(Z) = i\right] \\
&\geq \Pr\left[\text{all rows of } Z \text{ belong to } W \mid (b_1,\ldots,b_{k+1}) = 1^{k+1} \text{ and } \mathsf{rk}(Z) = i\right] \\
&= 1 - \Pr\left[\exists \text{ a row of } Z \text{ that } \notin W \mid (b_1,\ldots,b_{k+1}) = 1^{k+1} \text{ and } \mathsf{rk}(Z) = i\right] \\
&\geq 1 - (k+1) \cdot \Pr_{z\sim Q}[z \notin W] \\
&\geq 1 - \frac{\tau_i}{4} \\
&\geq 1 - o_c(1),
\end{aligned}$$

where the first inequality uses the assumption that bit j in the binary representation of W is 1 and the facts that the dimension of W is equal to i and that we are conditioning on $\mathsf{rk}[Z] = i$. The second inequality follows from the union bound, the third inequality follows from the assumption given in Lemma 2, and the last inequality follows from the fact that for every $i \in [k+1]$, we have that $\tau_i = o_c(1)$.

To show (iii) above, note that

$$\begin{aligned}
\overline{u}_{i,j} &= \overline{v}_i \cdot \frac{\overline{u}_{i,j}}{\overline{v}_i} \\
&\geq \omega_c(\tau) \cdot (1 - o_c(1)) \\
&\geq \omega_c(\tau),
\end{aligned}$$

where the first inequality follows from (i) and (ii) above.

We now turn to the (slightly different) case where bit j of W is equal to 0, and prove that in the execution of Algorithm 2, we will have that $u_{i,j}/v_i = o_c(1)$. Note that for any positive real numbers N, D and τ such that $\tau = o(D)$, we have that

$$\frac{N+\tau}{D-\tau} \leq \frac{N}{D} \cdot (1 + o(1)) + o(1).$$

Thus, it is enough to use the fact that $\tau = o_c(\overline{v}_i)$ (proven in (i) above) and to show the next statement:

(iv) if bit j of W is 0, then $(\overline{u}_{i,j}/\overline{v}_i) = o_c(1)$.

To prove (iv), note that since bit j of W is 0, we have that

$$\begin{aligned}
\frac{\overline{u}_{i,j}}{\overline{v}_i} &\leq \Pr\left[\exists \text{ a row of } Z \text{ that } \notin W \mid (b_1,\ldots,b_{k+1}) = 1^{k+1} \text{ and } \mathsf{rk}(Z) = i\right] \\
&\leq \frac{\tau_i}{4} \\
&\leq o_c(1),
\end{aligned}$$

where the first inequality above follows from the assumption that bit j in the binary representation of W is 0 and the facts that the dimension of W is equal to i and that we are conditioning on $\mathsf{rk}[Z] = i$. The second inequality above follows from the union bound and the assumption given in Lemma 2, and the last inequality follows from the fact that for every $i \in [k+1]$, we have that $\tau_i = o_c(1)$. As before, we choose c to be sufficiently large to ensure that this last probability is smaller than $(1/10)$.

A.2 Proof of Proposition 3.12

Let $a \in \mathbb{F}_p^\ell$. We have that:

$$\begin{aligned}
\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)] &= \mathbb{E}_{(z,b)\sim D_0}\left[\prod_{i=1}^{k}\mathbb{E}_{(z_i,b_i)\sim D_0}[D_a(z_i,b_i)\right] \\
&= \prod_{i=1}^{k}\mathbb{E}_{(z_i,b_i)\sim D_0}\left[D_a(z_i,b_i)\right] \\
&= \prod_{i=1}^{k}\mathbb{E}_{(z_i,b_i)\sim D_0}\left[D_a(z_i)\cdot\mathbb{1}(b_i = f_a(z_i))\right] \\
&= \prod_{i=1}^{k}\mathbb{E}_{z_i\sim D_0}\left[D_a(z_i)\cdot\mathbb{E}_{b_i\in_R\{\pm 1\}}[\mathbb{1}(b_i = f_a(z_i))]\right] \\
&= \frac{1}{2^k}\cdot\prod_{i=1}^{k}\mathbb{E}_{z_i\sim D_0}\left[D_a(z_i)\right] \\
&= \frac{1}{2^k}\cdot\left(\frac{1}{p}\cdot\beta + \left(1-\frac{1}{p}\right)\cdot\alpha\right)^k.
\end{aligned}$$

A.3 Proof of Proposition 3.13

Let $a, a' \in \mathbb{F}_p^\ell$. First, assume that $\mathsf{Hyp}_a = \mathsf{Hyp}_{a'}$, i.e., that $a = a'$. Then,

$$\begin{aligned}
\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)\cdot D_{a'}(z,b)] &= \mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)^2] \\
&= \mathbb{E}_{(z,b)\sim D_0}\left[\prod_{i=1}^{k}D_a(z_i,b_i)^2\right] \\
&= \prod_{i=1}^{k}\mathbb{E}_{(z_i,b_i)\sim D_0}[D_a(z_i,b_i)^2] \\
&= \prod_{i=1}^{k}\mathbb{E}_{(z_i,b_i)\sim D_0}[D_a(z_i)^2\cdot\mathbb{1}(b_i = f_a(z_i))] \\
&= \prod_{i=1}^{k}\mathbb{E}_{z_i}\left[D_a(z_i)^2\cdot\mathbb{E}_{b_i}[\mathbb{1}(b_i = f_a(z_i))]\right]
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)\cdot D_{a'}(z,b)] &= \frac{1}{2^k}\cdot\prod_{i=1}^{k}\mathbb{E}_{z_i}[D_a(z_i)^2] \\
&= \frac{1}{2^k}\cdot\prod_{i=1}^{k}\left(\frac{1}{p}\cdot\beta^2 + \left(1-\frac{1}{p}\right)\cdot\alpha^2\right) \\
&= \frac{1}{2^k}\cdot\left(\frac{1}{p}\cdot\beta^2 + \left(1-\frac{1}{p}\right)\cdot\alpha^2\right)^k.
\end{aligned}$$

Now we assume that $\mathsf{Hyp}_a \cap \mathsf{Hyp}_{a'} = \emptyset$. Then,

$$\begin{aligned}
\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)\cdot D_{a'}(z,b)] &= \mathbb{E}_{(z,b)\sim D_0}\left[\prod_{i=1}^{k} D_a(z_i,b_i)\cdot D_{a'}(z_i,b_i)\right] \\
&= \prod_{i=1}^{k} \mathbb{E}_{(z_i,b_i)\sim D_0}[D_a(z_i,b_i)\cdot D_{a'}(z_i,b_i)] \\
&= \prod_{i=1}^{k} \mathbb{E}_{(z_i,b_i)\sim D_0}[D_a(z_i)\cdot \mathbb{1}(b_i = f_a(z_i)) \\
&\quad \cdot D_{a'}(z_i)\cdot \mathbb{1}(b_i = f_{a'}(z_i))] \\
&= \prod_{i=1}^{k} \mathbb{E}_{z_i}\Big[D_a(z_i)\cdot D_{a'}(z_i)\cdot \mathbb{1}(f_a(z_i) = f_{a'}(z_i)) \\
&\quad \cdot \mathbb{E}_{b_i}[\mathbb{1}(b_i = f_a(z_i))]\Big] \\
&= \frac{1}{2^k}\cdot \prod_{i=1}^{k} \mathbb{E}_{z_i}\Big[D_a(z_i)\cdot D_{a'}(z_i)\cdot \mathbb{1}(f_a(z_i) = f_{a'}(z_i))\Big] \\
&= \frac{1}{2^k}\cdot \prod_{i=1}^{k} \left(\alpha^2\cdot\left(1-\frac{2}{p}\right)\right) \\
&= \frac{1}{2^k}\cdot \left(\alpha^2\cdot\left(1-\frac{2}{p}\right)\right)^k.
\end{aligned}$$

Finally, we assume that $\mathsf{Hyp}_a \neq \mathsf{Hyp}_{a'}$ and $\mathsf{Hyp}_a \cap \mathsf{Hyp}_{a'} \neq \emptyset$. Then,

$$\begin{aligned}
\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)\cdot D_{a'}(z,b)] &= \frac{1}{2^k}\cdot \prod_{i=1}^{k} \mathbb{E}_{z_i}\Big[D_a(z_i)\cdot D_{a'}(z_i)\cdot \mathbb{1}(f_a(z_i) = f_{a'}(z_i))\Big] \\
&= \frac{1}{2^k}\cdot \prod_{i=1}^{k} \left(\frac{\beta^2}{p^2} + \alpha^2\cdot\left(1-\frac{2}{p}+\frac{1}{p^2}\right)\right) \\
&= \frac{1}{2^k}\cdot \left(\frac{\beta^2}{p^2} + \alpha^2\cdot\left(1-\frac{2}{p}+\frac{1}{p^2}\right)\right)^k.
\end{aligned}$$

A.4 Proof of Proposition 3.14

First, we assume that $a, a' \in \mathbb{F}_p^\ell$ are such that $\mathsf{Hyp}_a = \mathsf{Hyp}_{a'}$, i.e., $a = a'$. Then, by Proposition 3.13 and by our settings of α and β, we have that

$$\begin{aligned}
\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)\cdot D_{a'}(z,b)] &= \frac{1}{2^k}\cdot \left(\frac{1}{p}\cdot\beta^2 + \left(1-\frac{1}{p}\right)\cdot\alpha^2\right)^k \\
&= \frac{1}{2^{2k}\cdot p^{(2\ell-1)\cdot k}}\cdot\left(1+\frac{1}{p-1}\right)^k.
\end{aligned}$$

Hence, $D_0[\hat{D}_a\cdot\hat{D}_{a'}] = \left(p+1-\frac{1}{p-1}\right)^k - 1$, as desired.

Next, we assume that $a, a' \in \mathbb{F}_p^\ell$ are such that $\mathsf{Hyp}_a \cap \mathsf{Hyp}_{a'} = \emptyset$. Then, by Proposition 3.13

and by our setting of α, we have that

$$\begin{aligned}\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)\cdot D_{a'}(z,b)] &= \frac{1}{2^k}\cdot(\alpha^2\cdot(1-\frac{2}{p}))^k \\ &= \frac{1}{2^{3k}\cdot p^{2k\ell}}\cdot\frac{(1-\frac{2}{p})^k}{(1-\frac{1}{p})^{2k}}.\end{aligned}$$

Hence, $D_0[\hat{D}_a\cdot\hat{D}_{a'}] = \frac{1}{2^k}\cdot\frac{(1-\frac{2}{p})^k}{(1-\frac{1}{p})^{2k}} - 1$, as desired.

Finally, we assume that $a, a' \in \mathbb{F}_p^\ell$ are such that $\mathsf{Hyp}_a \neq \mathsf{Hyp}_{a'}$ and $\mathsf{Hyp}_a \cap \mathsf{Hyp}_{a'} \neq \emptyset$. Then, by Proposition 3.13 and by our settings of α and β, we have that

$$\begin{aligned}\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)\cdot D_{a'}(z,b)] &= \frac{1}{2^k}\cdot(\frac{\beta^2}{p^2}+\alpha^2\cdot(1-\frac{2}{p}+\frac{1}{p^2}))^k \\ &= \frac{1}{2^{2k}\cdot p^{2k\ell}}.\end{aligned}$$

Hence, $D_0[\hat{D}_a\cdot\hat{D}_{a'}] = 0$, as desired.

Detecting Communities Is Hard (And Counting Them Is Even Harder)*

Aviad Rubinstein

University of California at Berkeley, Berkeley, USA
aviad@eecs.berkeley.edu

Abstract

We consider the algorithmic problem of community detection in networks. Given an undirected friendship graph $G = (V, E)$, a subset $S \subseteq V$ is an (α, β)-community if:

- Every member of the community is friends with an α-fraction of the community;
- Every non-member is friends with at most a β-fraction of the community.

Arora et al [3] gave a quasi-polynomial time algorithm for enumerating all the (α, β)-communities for any constants $\alpha > \beta$.

Here, we prove that, assuming the Exponential Time Hypothesis (ETH), quasi-polynomial time is in fact necessary - and even for a much weaker approximation desideratum. Namely, distinguishing between:

- G contains an $(1, o(1))$-community; and
- G does not contain a $(\beta + o(1), \beta)$-community for any $\beta \in [0, 1]$.

We also prove that counting the number of $(1, o(1))$-communities requires quasi-polynomial time assuming the weaker #ETH.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases Community detection, stable communities, quasipolynomial time

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.42

1 Introduction

Identifying communities is a central graph-theoretic problem with important applications to sociology and marketing (when applied to social networks), biology and bioinformatics (when applied to protein interaction networks), and more (see e.g. Fortunato's classic survey [21]). Defining what exactly is a *community* remains an interesting problem on its own (see Arora et al [3] and Borgs et al [11] for excellent treatment from a theoretical perspective). Ultimately, there is no single "right" definition, and the precise meaning of community should be different for social networks and protein interaction networks.

In this paper we focus on the algorithmic questions arising from one of the simplest and most canonical definitions, which has been considered by several theoretical computer scientists [30, 3, 5, 12] (see Subsection 1.1 for further discussion):

▶ **Definition 1** ((α, β)-Community)**.** Given an undirected graph $G = (V, E)$ an (α, β)-community is a subset $S \subseteq V$ that satisfies:

Strong ties inside the community For every $v \in S$, $|\{v\} \times S| \cap E \geq \alpha \cdot |S|$; and

Weak ties to nodes outside the community For every $u \notin S$, $|\{u\} \times S| \cap E \leq \beta \cdot |S|$.

* This research was supported by a Microsoft Research PhD Fellowship, as well as NSF grant CCF1408635 and Templeton Foundation grant 3966. This work was done in part at the Simons Institute for the Theory of Computing.

© Aviad Rubinstein;
licensed under Creative Commons License CC-BY

Arora et al [3, Theorem 3.1] gave a simple quasi-polynomial ($n^{O(\log n)}$) time for detecting (α, β)-communities whenever $\alpha - \beta$ is at least some positive constant. The algorithm enumerates over $O(\log n)$-tuples of vertices. For each tuple, consider the set of vertices that are neighbors of an $(\alpha + \beta)/2$-fraction of the tuple; test whether this candidate set is indeed a community.

Arora et al's algorithm and analysis are very similar to related algorithms for approximate Nash equilibrium [27], Densest k-Subgraph [8] and Dughmi's Zero-Sum Signaling problem [17]. Recently, matching quasi-polynomial hardness results have been proved for approximate Nash equilibrium [13, 4, 35, 18], Densest k-Subgraph [12, 29], and Zero-Sum Signaling [34, 10] using or inspired by the technique of "birthday repetition" [1]. A natural question, made explicit in [12], is whether similar techniques can be shown to prove quasi-polynomial time hardness, assuming the Exponential Time Hypothesis (ETH)[1], for (α, β)-community detection, for any constants $\alpha > \beta \in [0, 1]$.

Here we show that, for *every* constants $\alpha > \beta \in (0, 1]$, community detection requires quasi-polynomial time (assuming ETH). For example, when $\alpha = 1$ and $\beta = 0.01$, this means that we can hide a clique C, such that every single vertex not in C is connected to at most 1% of C. Our main result is actually a much stronger inapproximability: even in the presence of a $(1, o(1))$-community, finding any $(\beta + o(1), \beta)$-community is hard.

▶ **Theorem 2.** *For every n there exists an $\epsilon = \epsilon(n) = o(1)$ such that, assuming ETH, distinguishing between the following requires time $n^{\tilde{\Omega}(\log n)}$:*
Completeness *G contains an $(1, \epsilon)$-community; and*
Soundness *G does not contain an $(\beta + \epsilon, \beta)$-community for any $\beta \in [0, 1]$.*

Unlike all quasi-polynomial approximation schemes mentioned above, Arora et al's algorithm has the unique property that it can also *exactly count* all the (α, β)-communities. Our second result is that counting even the number of $(1, o(1))$-communities requires quasi-polynomial time. A nice feature of this result is that we can base it on the much weaker #ETH assumption, which asserts that counting the satisfying assignment for a 3SAT instance requires time $2^{\Omega(n)}$. (Note, for example, that #ETH is likely to be true even if $\mathsf{P} = \mathsf{NP}$.)

▶ **Theorem 3.** *For every n there exists an $\epsilon = \epsilon(n) = o(1)$ such that, assuming #ETH, counting $(1, \epsilon)$-communities requires time $n^{\log^{1-o(1)} n}$.*

1.1 Related works

The most closely related work is a reduction by Balcan, Borgs, Braverman, Chayes, and Teng [5, Theorem 5.3] from Planted Clique to finding $(1, 1 - \gamma)$-communities, for some small (unspecified) constant $\gamma > 0$. Note that our inapproximability in Theorem 2 is much stronger in all parameters; furthermore, although formally incomparable, our ETH assumption is preferable over the average-case hardness assumption of Planted Clique.

Algorithms for special cases

Mishra, Schreiber, Stanton, and Tarjan [30] gave a polynomial-time algorithm for finding (α, β)-communities that contain a vertex with very few neighbors outside the community.

[1] The Exponential Time Hypothesis (ETH) [25] asserts that solving 3SAT requires time $2^{\Omega(n)}$. Note that (given our current understanding of complexity) this assumption is essentially necessary - an NP-hardness result is very unlikely given [3]'s quasi-polynomial algorithm. Recall also that ETH is a significantly weaker assumption than the related SETH [24, 14] and NSETH [15],

Balcan et al [5] give a polynomial-time algorithm for enumerating (α, β)-communities in the special case where the degree of every node is $\Omega(n)$.

Arora, Ge, Sachdeva, and Schoenebeck [3] consider several semi-random models where the edges inside the community are generated at random, according to the expected degree model. (In fact, their quasi-polynomial time algorithm is also stated in this setting, but only their "Gap Assumption", which is equivalent to $\alpha - \beta = \Omega(1)$, is used in the analysis.)

Stochastic Block Model

Variants of the community detection problem on graphs generated by different stochastic models are extremely popular (see e.g. [6, 7, 16, 20, 22, 28, 31, 33, 37] for papers in conference proceedings from June 2016). Perhaps the most influential is the *Stochastic Block Model* [23]: The graph is partitioned into two disjoint communities; the edges within each community are present with probability α, independently, whereas edges between communities are present with probability β. Hence this model can also be seen as a special case of the (α, β)-Community Detection problem.

Stochastic models are extremely helpful in physics, for example, because atoms' interactions obey simple mathematical formulas with high precision. Unfortunately, for applications such as social networks, existing models do not describe human behavior with atomic precision, hence casting a shadow over the applicability of algorithms that work on ideal stochastic models. Recent works [31, 28] attempted to bridge the gap from ideal model to practice by showing that certain SDP-based algorithms continue to work in a particular semi-random model where a restricted adversary is allowed to modify the random input graph. These success stories beg the question of how strong can one make the adversary? The current paper illuminates some of the computational barriers.

Alternative approaches to modeling communities

As we mentioned above, there are many different definitions of "communities" in networks. For in-depth discussion of different definitions see Arora et al [3] or Borgs et al [11]. As pointed out by the latter, for some definitions even verifying that a candidate subset is a community is intractable.

There is also an important literature on axiomatic approaches to the related problem of clustering (e.g. [26, 9, 38]); note that while clustering typically aims to partition a set of nodes, our main focus is on detecting just a single community; in particular, different communities may intersect.

1.2 Overview of proofs

A good starting point for the technical discussion is a recent subexponential reduction from 3SAT to the related problem of Densest-k-Subgraph [12]. In Densest-k-Subgraph, we seek a subgraph of size k of maximal density. The two ingredients in [12]'s reduction are "birthday repetition" [1] and the "FGLSS graph" [19]:

"Birthday repetition" Starting with an instance of Label Cover (see definition in Section 2), the reduction considers a mega-variable for every ρ-tuple of variables, for $\rho \approx \sqrt{n}$. By the birthday paradox, almost every pair of ρ-tuples of variables intersect, inducing a consistency constraint on the two mega-assignments. Similarly, we expect to see some Label Cover edges in the union of the two ρ-tuples, inducing an additional Label Cover constraint between the two mega assignments. Notice that we have $\binom{n}{\rho} \approx 2^{\sqrt{n}}$

mega variables, and the alphabet size is also approximately $N = 2^{\sqrt{n}}$. Therefore, assuming ETH, finding an approximately satisfying assignment for the mega-variables requires time $2^{\Omega(n)} \approx N^{\log N}$.

FGLSS Similarly to the classic reduction by Feige et al. [19] for the Clique problem, [12] construct a vertex for each mega assignment to each mega variable, and draw an edge between two vertices if the induced assignments do not violate any consistency or Label Cover constraints. Notice that if the Label Cover instance has a satisfying assignment, then the graph contains a clique of size $\binom{n}{\rho}$ where each mega variable receives the mega assignment induced by the globally satisfying assignment. On the other hand, any subgraph that corresponds to a consistent assignment which violates many constraints must be missing most of its edges.

Now this simple reduction is still far from working for the Community Detection problem, and indeed the latter was listed as an open problem in [13]. Below we describe some of the obstacles and outline how we overcome them.

Completeness

Surprisingly, the main problem with using the same reduction for Community Detection is the completeness: even if the Label Cover instance has a satisfying assignment, the resulting graph has no (α, β)-communities, for any constants $\alpha > \beta$! Observe, in particular, that the clique that corresponds to the satisfying assignment does not satisfy the weak ties condition. For any vertex v in that clique, consider any vertex v' that corresponds to changing the assignment to just one variable x_i in v's assignment. If v agrees with the assignments of all other vertices in the clique, v' agrees with almost all of them - except for the negligible fraction that cover x_i or its neighbors in the Label Cover graph.

To overcome this problem of vertices that are "just outside the community", we use error correcting codes. Namely, we encode each assignment as a low-degree bivariate polynomial over finite field $\mathcal{G}$ of size $|\mathcal{G}| \approx \sqrt{n}$. Now vertices correspond to low-degree assignments to rows/columns of the polynomial. This guarantees that the assignments induced by every two vertices are far. If v agrees with all other vertices in the community, then almost all of those vertices disagree with v'.

Soundness

The main challenge for soundness is ruling out communities that do not correspond to a single, globally consistent assignment to the Label Cover instance. The key idea is to introduce auxiliary vertices that punish such communities by violating the weak ties desideratum.

Let us begin with the reduction to the counting variant (Theorem 3), which is easier, mostly because we are not concerned with approximation (i.e. we only have to show that subsets that are exactly $(1, \epsilon)$-communities correspond to satisfying assignments). Here we further simplify matters by sketching a construction with weighted edges. The full reduction (Section 3) uses unweighted edges and is only slightly more involved. Consider, for every $g \in \mathcal{G}$, an auxiliary vertex that is ϵ-connected to all proper vertices that do not correspond to assignments to the g-th row/column. Now if a $(1, \epsilon)$-community C does not contain a vertex with assignment to the g-th row/column, the auxiliary vertex must simultaneously: (i) belong to C so as not to violate the weak ties desideratum; yet (ii) it cannot belong to C because all its edges have weight ϵ (this would violate the strong ties desideratum). Therefore every $(1, \epsilon)$-community assigns values to every row/column in $\mathcal{G}^2$.

The reduction we described above suffices to show that (assuming ETH) deciding whether the graph contains a $(1, \epsilon)$-community also requires quasi-polynomial time. To get the stronger statement of Theorem 2 we must rule out even $(\beta, \beta + \epsilon)$-communities in case the Label Cover instance is far from satisfiable. In particular, we need to show that subsets that do not correspond to unique, consistent assignments are never $(\beta, \beta + \epsilon)$-communities. Instead of a single column/row, we let each proper vertex correspond to a subset of $t \approx \log n$ columns/rows. Instead of a single $g \in \mathcal{G}$, each auxiliary vertex corresponds to subset $H \subset \mathcal{G}$ of size $|H| = |\mathcal{G}|/2$. We draw an edge between an auxiliary vertex and a proper vertex if the indices of all t columns/rows are contained in H; if they are picked randomly this only happens with polynomially small probability. If, however, a β-fraction of the community is restricted to a small subset $R \subset \mathcal{G}$, then there are auxiliary vertices for $H \supseteq R$ that connect to all those nodes and violate the weak ties desideratum. Roughly, we show that at least a $(1 - \beta)$-fraction of the vertices have assignments that are "well spread" over $\mathcal{G}^2$, and among those assignments there are many violations of the Label Cover constraints.

2 Preliminaries

2.1 Label Cover

▶ **Definition 4** (Label Cover)**.** Label Cover is a maximization problem. The input is a bipartite graph $G = (A, B, E)$, alphabets Σ_A, Σ_B, and a projection $\pi_e : \Sigma_A \to \Sigma_B$ for every $e \in E$.

The output is a labeling $\varphi_A : A \to \Sigma_A$, $\varphi_B : B \to \Sigma_B$. Given a labeling, we say that a constraint (or edge) $(a, b) \in E$ is *satisfied* if $\pi_{(a,b)}(\varphi_A(a)) = \varphi_B(b)$. The *value of a labeling* is the fraction of $e \in E$ that are satisfied by the labeling. The value of the instance is the maximum fraction of constraints satisfied by any assignment.

▶ **Theorem 5** (Moshkovitz-Raz PCP [32, Theorem 11])**.** *For every n and every $\epsilon > 0$ (in particular, ϵ may be a function of n), solving* 3SAT *on inputs of size n can be reduced to distinguishing between the case that a (d_A, d_B)-bi-regular instance of* Label Cover*, with parameters $|A| + |B| = n^{1+o(1)} \cdot \mathsf{poly}(1/\epsilon)$, $|\Sigma_A| = 2^{\mathsf{poly}(1/\epsilon)}$, and $d_A, d_B, |\Sigma_B| = \mathsf{poly}(1/\epsilon)$, is completely satisfiable, versus the case that it has value at most ϵ.*

Counting the number of satisfying assignments is even harder. The following hardness is well-known, and we sketch its proof only for completeness:

▶ **Fact 1.** *There is a linear-time reduction from #3SAT to counting the number of satisfying assignments of a* Label Cover *instance.*

Proof. Construct a vertex in A for each variable and a vertex in B for each clause. Set $\Sigma_A \triangleq \{0, 1\}$ and let $\Sigma_B \triangleq \{0, 1\}^3 \setminus (000)$ (i.e. Σ_B is the set of satisfying assignments for a 3SAT clause, after applying negations). Now if variable x appears in clause C, add a constraint that the assignments to x and C are consistent (taking into account the sign of x in C). Notice that any assignment to A: (i) corresponds to a unique assignment to the 3SAT formula; and (ii) if the 3SAT formula is satisfied, this assignment uniquely defines a satisfying assignment to B. Therefore there is a one-to-one correspondence between satisfying assignments to the 3SAT formula and to the instance of Label Cover. ◀

2.2 Finding a good partition

▶ **Theorem 6** (k-wise independence Chernoff bound [36, Theorem 5.I]). *Let $x_1 \dots x_n \in [0,1]$ be k-wise independent random variables, and let $\mu \triangleq \mathsf{E}\left[\sum_{i=1}^{n} x_i\right]$ and $\delta \leq 1$. Then*

$$\Pr\left[\left|\sum_{i=1}^{n} x_i - \mu\right| > \delta\mu\right] \leq \mathrm{e}^{-\Omega\left(\min\left\{k,\delta^2\mu\right\}\right)}.$$

We use Chernoff bound with $\Theta(\log n)$-wise independent variables to deterministically partition variables into subsets of cardinality $\approx \sqrt{n}$. Our (somewhat naive) deterministic algorithm for finding a good partition takes quasi-polynomial time ($n^{O(\log n)}$), which is negligible with respect to the sub-exponential size ($N = 2^{\tilde{O}(\sqrt{n})}$) of our reduction[2].

▶ **Lemma 7.** *Let $G = (A, B, E)$ be a bipartite (d_A, d_B)-bi-regular graph, and let $n_A \triangleq |A|$, $n_B \triangleq |B|$; set also $n \triangleq n_B + n_A$ and $\rho \triangleq \sqrt{n}\log n$. Let $T_1, \dots, T_{n_B/\rho}$ be an arbitrary partition of B into disjoint subsets of size ρ. There is a quasi-polynomial deterministic algorithm (alternatively, linear-time randomized algorithm) that finds a partition of A into $S_1, \dots, S_{n_A/\rho}$, such that:*

$$\forall i \quad \left||S_i| - \rho\right| < \rho/2, \tag{1}$$

and

$$\forall i,j \quad \left|\left|(S_i \times T_j) \cap E\right| - \frac{d_A\rho^2}{n_B}\right| < \frac{d_A\rho^2}{2n_B}. \tag{2}$$

Proof. Suppose that we place each $a \in A$ into a uniformly random S_i. By Chernoff bound and union bound, (1) and (2) hold with high probability. Now, by Chernoff Bound for k-wise independent variables (Theorem 6), it suffices to partition A using a $\Theta(\log n)$-wise independent distribution. Such distribution can be generated with a sample space of $n^{O(\log n)}$ (e.g. [2]). Therefore, we can enumerate over all possibilities in quasi-polynomial time. By the probabilistic argument, we will find at least one partition that satisfies (1) and (2). ◀

3 Hardness of Counting Communities

▶ **Theorem 8.** *There exists an $\epsilon(n) = o(1)$ such that, assuming #ETH, counting $(1,\epsilon)$-communities requires time $n^{\log^{1-o(1)} n}$.*

Construction

Begin with an instance (A, B, E, π) of Label Cover of size $n = n_A + n_B$ where $n_A \triangleq |A|$ and $n_B \triangleq |B|$. Let $\mathcal{G}$ be a finite field of size $\sqrt{n}/\epsilon^3$, and let $\mathcal{F} \subset \mathcal{G}$ be an arbitrary subset of size $|\mathcal{F}| = \sqrt{n}$. We identify between $A \cup B$ and points in $\mathcal{F}^2$; we also identify between a subset of $\mathcal{G}$ and $\Sigma_A \cup \Sigma_B$. Thus there is a one-to-one correspondence between a subset of assignments to $P_{\mathcal{F}} \colon \mathcal{F}^2 \to \mathcal{G}$ and assignments to the Label Cover instance. We can extend any such $P_{\mathcal{F}}$ to an individual-degree-$(|\mathcal{F}|-1)$ polynomial $P : \mathcal{G}^2 \to \mathcal{G}$. In the other

[2] Do not confuse this with the quasi-polynomial lower bound ($N^{\tilde{O}(\log N)}$) we obtain for the running time of the community detection problem.

direction, we think of each low individual degrees polynomial $P : \mathcal{G}^2 \to \mathcal{G}$ as a (possibly invalid) assignment to the Label Cover instance.

For every $g \in \mathcal{G}$, and degree-$(|\mathcal{F}|-1)$ polynomials $p_1, p_2 : \mathcal{G} \to \mathcal{G}$ such that $p_1(g) = p_2(g)$, we construct $1/\epsilon$ vertices $\{v_{g,p_1,p_2,i}\}_{i=1}^{1/\epsilon} \subset V$ in the communities graph. Each vertex naturally induces an assignment (p_1, p_2) on $(\mathcal{G} \times \{g\}) \cup (\{g\} \times \mathcal{G})$. We draw an edge between two vertices in V if they agree on the intersection of their lines, and if their induced assignments satisfy all the Label Cover constraints.

For every $g \in \mathcal{G}$ and $i \in [1/\epsilon]$, we also add two identical auxiliary vertices $u_{g,i}$ which are connected to every $v_{g',p_1,p_2,i}$ for $g' \neq g$ (but not to each other).

Completeness

For each assignment to the Label Cover instance, we construct a $(1,\epsilon)$-community by taking the induced assignment $P_{\mathcal{F}} \colon \mathcal{F}^2 \to \mathcal{G}$ and extending it to an individual-degree-$(|\mathcal{F}|-1)$ polynomial $P : \mathcal{G}^2 \to \mathcal{G}$. Let C be all the vertices $v_{g,p_1,p_2,i}$ such that p_1, p_2 are the restrictions of P to $(\mathcal{G} \times \{g\}), (\{g\} \times \mathcal{G})$. This correspondence is one-to-one and we need to show that the resulting C is actually a $(1,\epsilon)$-community.

Because all the vertices correspond to a consistent satisfying assignment, C is a clique. Let $v_{g,q_1,q_2,i} \notin C$; wlog q_1 disagrees with the restriction of P to $(\mathcal{G} \times \{g\})$. Since both q_1 and the restriction of P are degree-$(|\mathcal{F}|-1)$ polynomials, they must disagree on all but at most $(|\mathcal{F}|-1)$ elements of $\mathcal{G}$. For all other $h \in \mathcal{G}$, the vertex $v_{g,q_1,q_2,i}$ does not share edges with any $v_{h,p_1,p_2,j} \in C$. Therefore, $v_{g,q_1,q_2,i}$ has edges to less than an $(|\mathcal{F}| / |\mathcal{G}|)$-fraction of vertices in C. Finally, every auxiliary vertex $u_{g,i}$ has edges to a $\frac{|\mathcal{G}|-1}{|\mathcal{G}|} \cdot \epsilon < \epsilon$-fraction of the vertices in ϵ. Therefore, C is a $(1,\epsilon)$-community.

3.1 Soundness

Structure of $(1,\epsilon)$-communities

▶ **Claim 1.** *Every $(1,\epsilon)$-community C contains exactly $1/\epsilon$ vertices $\{v_{g,p_1,p_2,i}\}_{i=1}^{1/\epsilon}$ for each g.*

Proof. First, observe that C cannot contain any auxiliary vertices: if C contains one copy of $u_{g,i}$, it must also contain the other; but they don't have an edge between them, so they cannot both belong to a $(1,\epsilon)$-community.

Now, assume by contradiction that for some $g \in \mathcal{G}$, C does not contain any vertices with assignments for $(\mathcal{G} \times \{g\}) \cup (\{g\} \times \mathcal{G})$. Then every vertex in C is connected to (both copies of) $u_{g,i}$, for some $i \in [1/\epsilon]$. Therefore there is at least one $i \in [1/\epsilon]$ such that $u_{g,i}$ is connected to an ϵ-fraction of the vertices in C. But this is a contradiction since $u_{g,i} \notin C$.

If we ignore the auxiliary vertices (which, as we argued, C does not contain), the different vertices $v_{g,p_1,p_2,i}$ that correspond to the same assignment to the same lines (i.e. if we only change i) are indistinguishable. Therefore if C contains one of them, it must contain all of them (hence, at least $1/\epsilon$ vertices for each g).

Finally, since C is a clique, it cannot contain vertices that disagree on any assignments. (In particular, it cannot contain more than $1/\epsilon$ vertices for each g.) ◀

Completing the proof

Proof of Soundness. By Claim 1, every $(1,\epsilon)$-community C contains exactly $1/\epsilon$ vertices $\{v_{g,p_1,p_2,i}\}_{i=1}^{1/\epsilon}$ for each g. Furthermore, since C is a clique, all the induced assignments agree on all the intersections. So every $(1,\epsilon)$-community corresponds to a unique consistent

assignment to the Label Cover instance. Finally, appealing again to the fact that C is a clique, this assignment must also satisfy all the Label Cover constraints. ◀

4 Hardness of Detecting Communities

▶ **Theorem 9.** *There exists an $\epsilon(n) = o(1)$ such that, assuming ETH, distinguishing between the following requires time $n^{\tilde{\Omega}(\log n)}$:*

Completeness *G contains an $(1, \epsilon)$-community; and*

Soundness *G does not contain an $(\beta + \epsilon, \beta)$-community for any $\beta \in [0, 1]$.*

The rest of this section is devoted to the proof of Theorem 9. Our starting point is the Label Cover of Moshkovitz-Raz (Theorem 5). We compose the birthday repetition technique of [1] with a bi-variate low-degree encoding. We then encode this as a graph a-la FGLSS [19]. We add auxiliary vertices to ensure that any $(\beta + \epsilon, \beta)$-community corresponds, approximately, to a uniform distribution over the variables.

Construction

Begin with a (d_A, d_B)-bi-regular instance (A, B, E, π) of Label Cover of size $n = n_A + n_B$ where $n_A \triangleq |A|$ and $n_B \triangleq |B|$. Let $\rho \triangleq \sqrt{n} \log n$; let $\mathcal{G}$ be a finite field of size $\rho/\epsilon^3 = \tilde{O}(\rho)$, and let $\mathcal{F} \subset \mathcal{G}$ be an arbitrary subset of size $|\mathcal{F}| = 2\rho$. Let $\mathcal{F}_A, \mathcal{F}_B \subset \mathcal{F}$ be disjoint subsets of size n_A/ρ, n_B/ρ, respectively. By Lemma 7, we can partition A and B into subsets $X_1, \ldots, X_{|\mathcal{F}_A|}$ and $Y_1, \ldots, Y_{|\mathcal{F}_B|}$ of size at most $|\mathcal{F}|$ such that between every two subsets there are approximately $\frac{d_A \rho^2}{n_B} = \frac{d_B \rho^2}{n_A}$ constraints. For $i \in \mathcal{F}_A$, we think of the points $\{i\} \times \mathcal{F} \subset \mathcal{G}^2$ as representing assignments to variables in X_i; for $j \in \mathcal{F}_B$, we think of $\mathcal{F} \times \{j\} \subset \mathcal{G}^2$ as representing assignments to variables in Y_j. Notice that each point in $\mathcal{F}^2$ may represent an assignment to both a vertex from A and a vertex from B, to one of them, or to neither. In particular, any assignment $P \colon \mathcal{G}^2 \to \mathcal{G}$ induces an assignment for the Label Cover instance; note that since $|\mathcal{G}| > |\Sigma_A| \, |\Sigma_B|$, one value $P(f_1, f_2) \in \mathcal{G}$ suffices to describe assignments to both $a \in A$ and $b \in B$.

Let $t \triangleq \log n \cdot \left(\frac{|\mathcal{G}|}{|\mathcal{F}_A|} + \frac{|\mathcal{G}|}{|\mathcal{F}_B|} \right) = \mathsf{polylog}(n)$. We say that a subset $S \in \binom{\mathcal{G}}{t}$ is *balanced* if: $|S \cap \mathcal{F}_A| = \frac{|\mathcal{F}_A|}{|\mathcal{G}|} \cdot t$ and $|S \cap \mathcal{F}_B| = \frac{|\mathcal{F}_B|}{|\mathcal{G}|} \cdot t$. For every balanced subset S, consider $2t$ polynomials $q_\ell \colon \mathcal{G} \to \mathcal{G}$ of degree at most $|\mathcal{F}| - 1$, representing an assignment[3] $Q \colon (S \times \mathcal{G}) \cup (\mathcal{G} \times S) \to \mathcal{G}$. For balanced S and $2t$-tuple of polynomials (q_ℓ), we construct a corresponding vertex $v_{S,(q_\ell)}$ in the communities graph. Let V denote the set of vertices defined so far. For $g \in \mathcal{G}$ we abuse notation and say that $g \in v_{S,(q_\ell)}$ if $g \in S$. We construct an edge in the communities graph between two vertices in V if their assignments agree on the variables in their intersection, and their induced assignments to $A \cup B$ satisfy all the Label Cover constraints.

Additionally, for every $H \subset \mathcal{G}$ of size $|H| = |\mathcal{G}|/2$, define $|V|^2$ identical auxiliary vertices u_H in the communities graph. We draw an edge between auxiliary vertex u_H and vertex $v_{S,(q_\ell)}$ if $S \subset H$. Similarly, for every $H_A \subset \mathcal{F}_A$ of size $|H_A| = |\mathcal{F}_A|/2$, we define $|V|^2$ identical auxiliary vertices u_{H_A} with edges to every vertex $v_{S,(q_\ell)}$ such that $(S \cap \mathcal{F}_A) \subset H_A$. For $H_B \subset \mathcal{F}_B$ of size $|H_B| = |\mathcal{F}_B|/2$, we draw edges between u_{H_B} and $v_{S,(q_\ell)}$ such that $(S \cap \mathcal{F}_B) \subset H_B$.

[3] We will only consider polynomials that correspond to a consistent assignment Q; i.e. for each point in $S \times S$ we expect the two corresponding polynomials to agree with each other.

Completeness

Suppose that the Label Cover instance has a satisfying assignment. Let $\mathcal{Z} \subseteq \mathcal{G}^2$ denote the subset of points that correspond to at least one variable in A or B. Let $P_{\mathcal{Z}} : \mathcal{Z} \to \mathcal{G}$ be the induced function on $\mathcal{Z}$ that corresponds to the satisfying assignment, and let $P : \mathcal{G}^2 \to \mathcal{G}$ be the extension of $P_{\mathcal{Z}}$ by setting $P(f_1, f_2) = 0$ for $(f_1, f_2) \in \mathcal{F}^2 \setminus \mathcal{Z}$ (this choice is arbitrary), and then extending to an $(|\mathcal{F}| - 1)$-individual-degree polynomial over all of $\mathcal{G}^2$.

Let C be the set of vertices that correspond to restrictions of P to balanced sets, i.e.

$$C = \left\{ v_{S,(P|_S)} : S \text{ is balanced} \right\},$$

where $P\ |_S$ denotes the restriction of P to $(S \times \mathcal{G}) \cup (\mathcal{G} \times S)$. Since all those vertices correspond to a consistent satisfying assignment, C is a clique.

For any vertex $v_{S,(q_\ell)} \notin C$, at least one of the polynomials, q_{ℓ^*} disagrees with the restriction of P to the corresponding line. Since both q_{ℓ^*} and the restriction of P to that line are degree-$(|\mathcal{F}| - 1)$ polynomials, they must disagree on at least $\left(1 - \frac{|\mathcal{F}|}{|\mathcal{G}|}\right)$-fraction of the coordinates. The probability that a random balanced set S' is contained in the $O\left(\epsilon^3\right)$-fraction of coordinates where they do agree is smaller than ϵ (and in fact polynomially small in n). Therefore $v_{S,(q_\ell)}$ has inconsistency violations with all but (less than) an ϵ-fraction of the vertices in C.

For any auxiliary vertex u_{H_A}, the probability that a random vertex $v_{S,(P|_S)} \in C$ is connected to u_{H_A} is $2^{-|S \cap \mathcal{F}_A|} < 1/n$, and similarly for u_{H_B} and u_H. Therefore, every auxiliary vertex is connected to less than a $(1/n)$-fraction of the vertices in C.

4.1 Soundness

▶ **Lemma 10.** *If the* Label Cover *instance has value at most ϵ^3, then there are no $(\beta + \epsilon, \beta)$-communities.*

4.1.0.1 Auxiliary vertices

▶ **Claim 2.** *Every $(\beta + \epsilon, \beta)$-community does not contain any auxiliary vertices.*

Proof. There are $|V|^2$ identical copies of each auxiliary vertex. Since they are identical, any community must either contain all of them, or none of them. If the community contains all $|V|^2$ copies, then it has a vast majority of auxiliary vertices, so none of them can have edges to an ϵ-fraction of the community. ◀

4.1.0.2 List decoding

▶ **Claim 3.** *The vertices in any $(\beta + \epsilon, \beta)$-community C induce at most $4/\epsilon$ different assignments for each variable.*

Proof. Suppose by contradiction that this is not the case. Then, wlog, there is a line $\{g_1\} \times \mathcal{G}$ that receives at least $2/\epsilon$ different assignments from vertices in C. Every two assignments agree on at most $|\mathcal{F}|$ points (g_1, g') on the line, so in total there are at most $2|\mathcal{F}|/\epsilon^2$ points where at least two assignments agree. Let $R \subseteq \mathcal{G}$ denote the set of g' such that no two assignments agree on (g_1, g'); we have that $|R| \geq |\mathcal{G}| - 2|\mathcal{F}|/\epsilon^2 \geq |\mathcal{G}|/2$. Therefore, by the weak ties property, for at most a β-fraction of the vertices $v_{S,(q_\ell)} \in C$, $S \cap R = \emptyset$.

Consider the remaining $(1 - \beta)$-fraction of vertices in C. Suppose that v assigns a value to some (g_1, g') for $g' \in R$: this value can only agree with one of the $2/\epsilon$ different assignments

to (g_1, g'). Therefore, in expectation, each of the $2/\epsilon$ vertices that assign different values for (g_1, g') is connected to at most a $(\beta + \epsilon/2)$-fraction of the vertices in C. This is a contradiction to C being a $(\beta + \epsilon, \beta)$-community. ◀

4.1.0.3 Completing the proof

Proof of Lemma 10. Suppose that at most a ϵ^3-fraction of the Label Cover constraints can be satisfied by any single assignment, and assume by contradiction that C is a $(\beta + \epsilon, \beta)$-community. By Claim 3, C induces at most $4/\epsilon$ assignments on each variable, so at most $O(\epsilon)$-fraction of the constraints are satisfied by any pair of assignments.

By Markov's inequality, for at least half of the subsets $X_i \subset A$, only an $O(\epsilon)$-fraction of the constraints that depend on X_i are satisfied. By Claim 2 at least $(1 - \beta)$-fraction of the vertices in C assign values to at least one such X_i. Consider any such vertex $v_{S,(q_\ell)}$ where $S \ni i$. By construction of the partitions (Lemma 7), each X_i shares approximately the same number of constraints with each Y_j. Therefore, for all but an $O(\epsilon)$-fraction of Y_j's, X_i and Y_j observe a violation - for all the assignments given by vertices in C to the variables in Y_j. In other words, $v_{S,(q_\ell)}$ cannot have edges to any vertex $v_{T,(r_\ell)}$ such that $T \ni j$, for a $(1 - O(\epsilon))$-fraction of $j \in [n_B/k_B]$. Finally, applying Claim 2 again, at most a β fraction of vertices in C do not contain any of those j's. This is a contradiction to $v_{S,(q_\ell)}$ having edges to $(\beta + \epsilon)$-fraction of the vertices in C. ◀

References

1 Scott Aaronson, Russell Impagliazzo, and Dana Moshkovitz. AM with multiple merlins. In *Computational Complexity (CCC), 2014 IEEE 29th Conference on*, pages 44–55. IEEE, 2014.

2 Noga Alon, László Babai, and Alon Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *J. Algorithms*, 7(4):567–583, 1986. doi:10.1016/0196-6774(86)90019-2.

3 Sanjeev Arora, Rong Ge, Sushant Sachdeva, and Grant Schoenebeck. Finding overlapping communities in social networks: toward a rigorous approach. In *ACM Conference on Electronic Commerce, EC '12, Valencia, Spain, June 4-8, 2012*, pages 37–54, 2012. doi:10.1145/2229012.2229020.

4 Yakov Babichenko, Christos H. Papadimitriou, and Aviad Rubinstein. Can almost everybody be almost happy? In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 1–9, 2016. doi:10.1145/2840728.2840731.

5 Maria-Florina Balcan, Christian Borgs, Mark Braverman, Jennifer T. Chayes, and Shang-Hua Teng. Finding endogenously formed communities. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 767–783, 2013. doi:10.1137/1.9781611973105.55.

6 Afonso S. Bandeira, Nicolas Boumal, and Vladislav Voroninski. On the low-rank approach for semidefinite programs arising in synchronization and community detection. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 361–382, 2016. URL: http://jmlr.org/proceedings/papers/v49/bandeira16.html.

7 Jess Banks, Cristopher Moore, Joe Neeman, and Praneeth Netrapalli. Information-theoretic thresholds for community detection in sparse networks. In *Proceedings of the 29th Confer-*

ence on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016, pages 383–416, 2016. URL: http://jmlr.org/proceedings/papers/v49/banks16.html.

8 Siddharth Barman. Approximating nash equilibria and dense bipartite subgraphs via an approximate version of caratheodory's theorem. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 361–369, 2015. doi:10.1145/2746539.2746566.

9 Shai Ben-David and Margareta Ackerman. Measures of clustering quality: A working set of axioms for clustering. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 121–128, 2008. URL: http://papers.nips.cc/paper/3491-measures-of-clustering-quality-a-working-set-of-axioms-for-clustering.

10 Umang Bhaskar, Yu Cheng, Young Kun Ko, and Chaitanya Swamy. Hardness results for signaling in bayesian zero-sum and network routing games. In *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*, pages 479–496, 2016. doi:10.1145/2940716.2940753.

11 Christian Borgs, Jennifer T. Chayes, Adrian Marple, and Shang-Hua Teng. An axiomatic approach to community detection. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 135–146, 2016. doi:10.1145/2840728.2840748.

12 Mark Braverman, Young Kun-Ko, Aviad Rubinstein, and Omri Weinstein. ETH hardness for densest-k-subgraph with perfect completeness. In *SODA*, 2017. To appear.

13 Mark Braverman, Young Kun-Ko, and Omri Weinstein. Approximating the best nash equilibrium in $n^{o(\log n)}$-time breaks the exponential time hypothesis. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 970–982, 2015. doi:10.1137/1.9781611973730.66.

14 Chris Calabro, Russell Impagliazzo, and Ramamohan Paturi. The complexity of satisfiability of small depth circuits. In *Parameterized and Exact Computation, 4th International Workshop, IWPEC 2009, Copenhagen, Denmark, September 10-11, 2009, Revised Selected Papers*, pages 75–85, 2009. doi:10.1007/978-3-642-11269-0_6.

15 Marco L. Carmosino, Jiawei Gao, Russell Impagliazzo, Ivan Mihajlin, Ramamohan Paturi, and Stefan Schneider. Nondeterministic extensions of the strong exponential time hypothesis and consequences for non-reducibility. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 261–270, 2016. doi:10.1145/2840728.2840746.

16 Yuxin Chen, Govinda M. Kamath, Changho Suh, and David Tse. Community recovery in graphs with locality. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 689–698, 2016. URL: http://jmlr.org/proceedings/papers/v48/chena16.html.

17 Yu Cheng, Ho Yee Cheung, Shaddin Dughmi, Ehsan Emamjomeh-Zadeh, Li Han, and Shang-Hua Teng. Mixture selection, mechanism design, and signaling. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1426–1445, 2015. doi:10.1109/FOCS.2015.91.

18 Argyrios Deligkas, John Fearnley, and Rahul Savani. Inapproximability results for approximate nash equilibria. *CoRR*, abs/1608.03574, 2016. URL: http://arxiv.org/abs/1608.03574.

19 Uriel Feige, Shafi Goldwasser, Laszlo Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM (JACM)*, 43(2):268–292, 1996.

20 Laura Florescu and Will Perkins. Spectral thresholds in the bipartite stochastic block model. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 943–959, 2016. URL: http://jmlr.org/proceedings/papers/v49/florescu16.html.

21 Santo Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.

22 Bruce E. Hajek, Yihong Wu, and Jiaming Xu. Semidefinite programs for exact recovery of a hidden community. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 1051–1095, 2016. URL: http://jmlr.org/proceedings/papers/v49/hajek16.html.

23 Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5:109–137, 1983.

24 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

25 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

26 Jon M. Kleinberg. An impossibility theorem for clustering. In *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*, pages 446–453, 2002. URL: http://papers.nips.cc/paper/2340-an-impossibility-theorem-for-clustering.

27 Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *EC*, pages 36–41, 2003.

28 Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Learning communities in the presence of errors. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 1258–1291, 2016. URL: http://jmlr.org/proceedings/papers/v49/makarychev16.html.

29 Pasin Manurangsi. Almost-Polynomial Ratio ETH-Hardness of Approximating Densest k-Subgraph with Perfect Completeness. *CoRR*, abs/1611.05991, 2016.

30 Nina Mishra, Robert Schreiber, Isabelle Stanton, and Robert Endre Tarjan. Clustering social networks. In *Algorithms and Models for the Web-Graph, 5th International Workshop, WAW 2007, San Diego, CA, USA, December 11-12, 2007, Proceedings*, pages 56–67, 2007. doi:10.1007/978-3-540-77004-6_5.

31 Ankur Moitra, William Perry, and Alexander S. Wein. How robust are reconstruction thresholds for community detection? In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 828–841, 2016. doi:10.1145/2897518.2897573.

32 Dana Moshkovitz and Ran Raz. Two-query PCP with subconstant error. *J. ACM*, 57(5), 2010. doi:10.1145/1754399.1754402.

33 Elchanan Mossel and Jiaming Xu. Density evolution in the degree-correlated stochastic block model. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 1319–1356, 2016. URL: http://jmlr.org/proceedings/papers/v49/mossel16.html.

34 Aviad Rubinstein. Eth-hardness for signaling in symmetric zero-sum games. *CoRR*, abs/1510.04991, 2015. URL: http://arxiv.org/abs/1510.04991.

35 Aviad Rubinstein. Settling the complexity of computing approximate two-player nash equilibria. In *To appear in FOCS*, 2016.

36 Jeanette P. Schmidt, Alan Siegel, and Aravind Srinivasan. Chernoff-hoeffding bounds for applications with limited independence. *SIAM J. Discrete Math.*, 8(2):223–250, 1995. doi:10.1137/S089548019223872X.

37 Nicolas Tremblay, Gilles Puy, Rémi Gribonval, and Pierre Vandergheynst. Compressive spectral clustering. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 1002–1011, 2016. URL: `http://jmlr.org/proceedings/papers/v48/tremblay16.html`.

38 Twan van Laarhoven and Elena Marchiori. Axioms for graph clustering quality functions. *Journal of Machine Learning Research*, 15(1):193–215, 2014. URL: `http://dl.acm.org/citation.cfm?id=2627441`.

Inherent Trade-Offs in the Fair Determination of Risk Scores

Jon Kleinberg[1], Sendhil Mullainathan[2], and Manish Raghavan[3]

1 **Cornell University, Ithaca, USA**
kleinber@cs.cornell.edu
2 **Harvard University, Cambridge, USA**
mullain@fas.harvard.edu
3 **Cornell University, Ithaca, USA**
manish@cs.cornell.edu

Abstract

Recent discussion in the public sphere about algorithmic classification has involved tension between competing notions of what it means for a probabilistic classification to be fair to different groups. We formalize three fairness conditions that lie at the heart of these debates, and we prove that except in highly constrained special cases, there is no method that can satisfy these three conditions simultaneously. Moreover, even satisfying all three conditions approximately requires that the data lie in an approximate version of one of the constrained special cases identified by our theorem. These results suggest some of the ways in which key notions of fairness are incompatible with each other, and hence provide a framework for thinking about the trade-offs between them.

1998 ACM Subject Classification H.2.8 Database Applications, J.1 Administrative Data Processing

Keywords and phrases algorithmic fairness, risk tools, calibration

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.43

1 Introduction

There are many settings in which a sequence of people comes before a decision-maker, who must make a judgment about each based on some observable set of features. Across a range of applications, these judgments are being carried out by an increasingly wide spectrum of approaches ranging from human expertise to algorithmic and statistical frameworks, as well as various combinations of these approaches.

Along with these developments, a growing line of work has asked how we should reason about issues of bias and discrimination in settings where these algorithmic and statistical techniques, trained on large datasets of past instances, play a significant role in the outcome. Let us consider three examples where such issues arise, both to illustrate the range of relevant contexts, and to surface some of the challenges.

A set of example domains

First, at various points in the criminal justice system, including decisions about bail, sentencing, or parole, an officer of the court may use quantitative *risk tools* to assess a defendant's probability of recidivism — future arrest — based on their past history and other attributes. Several recent analyses have asked whether such tools are mitigating or exacerbating the sources of bias in the criminal justice system; in one widely-publicized report, Angwin et al.


© Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan;
licensed under Creative Commons License CC-BY

analyzed a commonly used statistical method for assigning risk scores in the criminal justice system — the COMPAS risk tool — and argued that it was biased against African-American defendants [1, 15]. One of their main contentions was that the tool's errors were asymmetric: African-American defendants were more likely to be incorrectly labeled as higher-risk than they actually were, while white defendants were more likely to be incorrectly labeled as lower-risk than they actually were. Subsequent analyses raised methodological objections to this report, and also observed that despite the COMPAS risk tool's errors, its estimates of the probability of recidivism are equally well calibrated to the true outcomes for both African-American and white defendants [17, 6, 9, 12].

Second, in a very different domain, researchers have begun to analyze the ways in which different genders and racial groups experience advertising and commercial content on the Internet differently [5, 18]. We could ask, for example: if a male user and female user are equally interested in a particular product, does it follow that they're equally likely to be shown an ad for it? Sometimes this concern may have broader implications, for example if women in aggregate are shown ads for lower-paying jobs. Other times, it may represent a clash with a user's leisure interests: if a female user interacting with an advertising platform is interested in an activity that tends to have a male-dominated viewership, like professional football, is the platform as likely to show her an ad for football as it is to show such an ad to an interested male user?

A third domain, again quite different from the previous two, is medical testing and diagnosis. Doctors making decisions about a patient's treatment may rely on tests providing probability estimates for different diseases and conditions. Here too we can ask whether such decision-making is being applied uniformly across different groups of patients [11, 19], and in particular how medical tests may play a differential role for conditions that vary widely in frequency between these groups.

Providing guarantees for decision procedures

One can raise analogous questions in many other domains of fundamental importance, including decisions about hiring, lending, or school admissions [16], but we will focus on the three examples above for the purposes of this discussion. In these three example domains, a few structural commonalities stand out. First, the algorithmic estimates are often being used as "input" to a larger framework that makes the overall decision — a risk score provided to a human expert in the legal and medical instances, and the output of a machine-learning algorithm provided to a larger advertising platform in the case of Internet ads. Second, the underlying task is generally about classifying whether people possess some relevant property: recidivism, a medical condition, or interest in a product. We will refer to people as being *positive instances* if they truly possess the property, and *negative instances* if they do not. Finally, the algorithmic estimates being provided for these questions are generally not pure yes-no decisions, but instead probability estimates about whether people constitute positive or negative instances.

Let us suppose that we are concerned about how our decision procedure might operate differentially between two groups of interest (such as African-American and white defendants, or male and female users of an advertising system). What sorts of guarantees should we ask for as protection against potential bias?

A first basic goal in this literature is that the probability estimates provided by the algorithm should be *well-calibrated*: if the algorithm identifies a set of people as having a probability x of constituting positive instances, then approximately an x fraction of this set should indeed be positive instances [4, 10]. Moreover, this condition should hold when

applied separately in each group as well [9]. For example, if we are thinking in terms of potential differences between outcomes for men and women, this means requiring that an x fraction of men and an x fraction of women assigned a probability x should possess the property in question.

A second goal focuses on the people who constitute positive instances (even if the algorithm can only imperfectly recognize them): the average score received by people constituting positive instances should be the same in each group. We could think of this as *balance for the positive class*, since a violation of it would mean that people constituting positive instances in one group receive consistently lower probability estimates than people constituting positive instances in another group. In our initial criminal justice example, for instance, one of the concerns raised was that white defendants who went on to commit future crimes were assigned risk scores corresponding to lower probability estimates in aggregate; this is a violation of the condition here. There is a completely analogous property with respect to negative instances, which we could call *balance for the negative class*. These balance conditions can be viewed as generalizations of the notions that both groups should have equal false negative and false positive rates.

It is important to note that balance for the positive and negative classes, as defined here, is distinct in crucial ways from the requirement that the average probability estimate globally over *all* members of the two groups be equal. This latter global requirement is a version of *statistical parity* [8, 3, 13, 14]. In some cases statistical parity is a central goal (and in some it is legally mandated), but the examples considered so far suggest that classification and risk assessment are much broader activities where statistical parity is often neither feasible nor desirable. Balance for the positive and negative classes, however, is a goal that can be discussed independently of statistical parity, since these two balance conditions simply ask that once we condition on the "correct" answer for a person, the chance of making a mistake on them should not depend on which group they belong to.

The present work: Trade-offs among the guarantees

Despite their different formulations, the calibration condition and the balance conditions for the positive and negative classes intuitively all seem to be asking for variants of the same general goal — that our probability estimates should have the same effectiveness regardless of group membership. One might therefore hope that it would be feasible to achieve all of them simultaneously.

Our main result, however, is that these conditions are in general incompatible with each other; they can only be simultaneously satisfied in certain highly constrained cases. Moreover, this incompatibility applies to *approximate* versions of the conditions as well.

In the remainder of this section we formulate this main result precisely, as a theorem building on a model that makes the discussion thus far more concrete.

1.1 Formulating the Goal

Let's start with some basic definitions. As above, we have a collection of people each of whom constitutes either a positive instance or a negative instance of the classification problem. We'll say that the *positive class* consists of the people who constitute positive instances, and the negative class consists of the people who constitute negative instances. For example, for criminal defendants, the positive class could consist of those defendants who will be arrested again within some fixed time window, and the negative class could consist of those who will not. The positive and negative classes thus represent the "correct" answer to the

classification problem; our decision procedure does not know them, but is trying to estimate them.

Feature vectors

Each person has an associated *feature vector* σ, representing the data that we know about them. Let p_σ denote the fraction of people with feature vector σ who belong to the positive class. Conceptually, we will picture that while there is variation within the set of people who have feature vector σ, this variation is invisible to whatever decision procedure we apply; all people with feature vector σ are indistinguishable to the procedure. Our model will assume that the value p_σ for each σ is known to the procedure.[1]

Groups

Each person also belongs to one of two *groups*, labeled 1 or 2, and we would like our decisions to be unbiased with respect to the members of these two groups.[2] In our examples, the two groups could correspond to different races or genders, or other cases where we want to look for the possibility of bias between them. The two groups have different distributions over feature vectors: a person of group t has a probability $a_{t\sigma}$ of exhibiting the feature vector σ. However, people of each group have the same probability p_σ of belonging to the positive class provided their feature vector is σ. In this respect, σ contains all the relevant information available to us about the person's future behavior; once we know σ, we do not get any additional information from knowing their group as well.[3]

Risk Assignments

We say that an *instance* of our problem is specified by the parameters above: a feature vector and a group for each person, with a value p_σ for each feature vector, and distributions $\{a_{t\sigma}\}$ giving the frequency of the feature vectors in each group.

Informally, risk assessments are ways of dividing people up into sets based on their feature vectors σ (potentially using randomization), and then assigning each set a probability estimate that the people in this set belong to the positive class. Thus, we define a *risk assignment* to consist of a set of "bins" (the sets), where each bin is labeled with a *score* v_b that we intend to use as the probability for everyone assigned to bin b. We then create a rule for assigning people to bins based on their feature vector σ; we allow the rule to divide people with a fixed feature vector σ across multiple bins (reflecting the possible use of randomization). Thus, the rule is specified by values $X_{\sigma b}$: a fraction $X_{\sigma b}$ of all people with feature vector σ are assigned to bin b. Note that the rule does not have access to the group t of the person being considered, only their feature vector σ. (As we will see, this does not mean that the rule is incapable of exhibiting bias between the two groups.) In summary, a

---

[1] Clearly the case in which the value of p_σ is unknown is an important version of the problem as well; however, since our main results establish strong limitations on what is achievable, these limitations are only stronger because they apply even to the case of known p_σ.

[2] We focus on the case of two groups for simplicity of exposition, but it is straightforward to extend all of our definitions to the case of more than two groups.

[3] As we will discuss in more detail below, the assumption that the group provides no additional information beyond σ does not restrict the generality of the model, since we can always consider instances in which people of different groups never have the same feature vector σ, and hence σ implicitly conveys perfect information about a person's group.

risk assignment is specified by a set of bins, a score for each bin, and values $X_{\sigma b}$ that define a mapping from people with feature vectors to bins.

Fairness Properties for Risk Assignments

Within the model, we now express the three conditions discussed at the outset, each reflecting a potentially different notion of what it means for the risk assignment to be "fair."

(A) *Calibration within groups* requires that for each group t, and each bin b with associated score v_b, the expected number of people from group t in b who belong to the positive class should be a v_b fraction of the expected number of people from group t assigned to b.

(B) *Balance for the negative class* requires that the average score assigned to people of group 1 who belong to the negative class should be the same as the average score assigned to people of group 2 who belong to the negative class. In other words, the assignment of scores shouldn't be systematically more inaccurate for negative instances in one group than the other.

(C) *Balance for the positive class* symmetrically requires that the average score assigned to people of group 1 who belong to the positive class should be the same as the average score assigned to people of group 2 who belong to the positive class.

Why Do These Conditions Correspond to Notions of Fairness?

All of these are natural conditions to impose on a risk assignment; and as indicated by the discussion above, all of them have been proposed as versions of fairness. The first one essentially asks that the scores mean what they claim to mean, even when considered separately in each group. The second and third ask that if two individuals in different groups exhibit comparable future behavior (negative or positive), they should be treated comparably by the procedure. In other words, a violation of, say, the second condition would correspond to the members of the negative class in one group receiving consistently higher scores than the members of the negative class in the other group, despite the fact that the members of the negative class in the higher-scoring group have done nothing to warrant these higher scores.

We can also interpret some of the prior work around our earlier examples through the lens of these conditions. For example, in the analysis of the COMPAS risk tool for criminal defendants, the critique by Angwin et al. focused on the risk tool's violation of conditions (B) and (C); the counter-arguments established that it satisfies condition (A). While it is clearly crucial for a risk tool to satisfy (A), it may still be important to know that it violates (B) and (C). Similarly, to think in terms of the example of Internet advertising, with male and female users as the two groups, condition (A) as before requires that our estimates of ad-click probability mean the same thing in aggregate for men and women. Conditions (B) and (C) are distinct; condition (C), for example, says that a female user who genuinely wants to see a given ad should be assigned the same probability as a male user who wants to see the ad.

1.2 Determining What is Achievable: A Characterization Theorem

When can conditions (A), (B), and (C) be simultaneously achieved? We begin with two simple cases where it's possible.

- *Perfect prediction.* Suppose that for each feature vector σ, we have either $p_\sigma = 0$ or $p_\sigma = 1$. This means that we can achieve perfect prediction, since we know each person's class label (positive or negative) for certain. In this case, we can assign all feature vectors σ with $p_\sigma = 0$ to a bin b with score $v_b = 0$, and all σ with $p_\sigma = 1$ to a bin b' with score $v_{b'} = 1$. It is easy to check that all three of the conditions (A), (B), and (C) are satisfied by this risk assignment.
- *Equal base rates.* Suppose, alternately, that the two groups have the same fraction of members in the positive class; that is, the average value of p_σ is the same for the members of group 1 and group 2. (We can refer to this as the *base rate* of the group with respect to the classification problem.) In this case, we can create a single bin b with score equal to this average value of p_σ, and we can assign everyone to bin b. While this is not a particularly informative risk assignment, it is again easy to check that it satisfies fairness conditions (A), (B), and (C).

Our first main result establishes that these are in fact the only two cases in which a risk assignment can achieve all three fairness guarantees simultaneously.

▶ **Theorem 1.** *Consider an instance of the problem in which there is a risk assignment satisfying fairness conditions (A), (B), and (C). Then the instance must either allow for perfect prediction (with p_σ equal to 0 or 1 for all σ) or have equal base rates.*

Thus, in every instance that is more complex than the two cases noted above, there will be some natural fairness condition that is violated by any risk assignment. Moreover, note that this result applies regardless of how the risk assignment is computed; since our framework considers risk assignments to be arbitrary functions from feature vectors to bins labeled with probability estimates, it applies independently of the method — algorithmic or otherwise — that is used to construct the risk assignment.

The conclusions of the first theorem can be relaxed in a continuous fashion when the fairness conditions are only approximate. In particular, for any $\varepsilon > 0$ we can define ε-approximate versions of each of conditions (A), (B), and (C) (specified precisely in the next section), each of which requires that the corresponding equalities between groups hold only to within an error of ε. For any $\delta > 0$, we can also define a δ-approximate version of the equal base rates condition (requiring that the base rates of the two groups be within an additive δ of each other) and a δ-approximate version of the perfect prediction condition (requiring that in each group, the average of the expected scores assigned to members of the positive class is at least $1 - \delta$; by the calibration condition, this can be shown to imply a complementary bound on the average of the expected scores assigned to members of the negative class).

In these terms, our approximate version of Theorem 1 is the following.

▶ **Theorem 2.** *There is a continuous function f, with $f(x)$ going to 0 as x goes to 0, so that the following holds. For all $\varepsilon > 0$, and any instance of the problem with a risk assignment satisfying the ε-approximate versions of fairness conditions (A), (B), and (C), the instance must satisfy either the $f(\varepsilon)$-approximate version of perfect prediction or the $f(\varepsilon)$-approximate version of equal base rates.*

Thus, anything that approximately satisfies the fairness constraints must approximately look like one of the two simple cases identified above.

Finally, in connection to Theorem 1, we note that when the two groups have equal base rates, then one can ask for the most accurate risk assignment that satisfies all three fairness conditions (A), (B), and (C) simultaneously. Since the risk assignment that gives the same

score to everyone satisfies the three conditions, we know that at least one such risk assignment exists; hence, it is natural to seek to optimize over the set of all such assignments. We consider this algorithmic question in the final technical section of the paper.

To reflect a bit further on our main theorems and what they suggest, we note that our intention in the present work isn't to make a recommendation on how conflicts between different definitions of fairness should be handled. Nor is our intention to analyze which definitions of fairness are violated in particular applications or datasets. Rather, our point is to establish certain unavoidable trade-offs between the definitions, regardless of the specific context and regardless of the method used to compute risk scores. Since each of the definitions reflect (and have been proposed as) natural notions of what it should mean for a risk score to be fair, these trade-offs suggest a striking implication: that outside of narrowly delineated cases, any assignment of risk scores can in principle be subject to natural criticisms on the grounds of bias. This is equally true whether the risk score is determined by an algorithm or by a system of human decision-makers.

Special Cases of the Model

Our main results, which place strong restrictions on when the three fairness conditions can be simultaneously satisfied, have more power when the underlying model of the input is more general, since it means that the restrictions implied by the theorems apply in greater generality. However, it is also useful to note certain special cases of our model, obtained by limiting the flexibility of certain parameters in intuitive ways. The point is that our results apply *a fortiori* to these more limited special cases.

First, we have already observed one natural special case of our model: cases in which, for each feature vector σ, only members of one group (but not the other) can exhibit σ. This means that σ contains perfect information about group membership, and so it corresponds to instances in which risk assignments would have the potential to use knowledge of an individual's group membership. Note that we can convert any instance of our problem into a new instance that belongs to this special case as follows. For each feature vector σ, we create two new feature vectors $\sigma^{(1)}$ and $\sigma^{(2)}$; then, for each member of group 1 who had feature vector σ, we assign them $\sigma^{(1)}$, and for each member of group 2 who had feature vector σ, we assign them $\sigma^{(2)}$. The resulting instance has the property that each feature vector is associated with members of only one group, but it preserves the essential aspects of the original instance in other respects.

Second, we allow risk assignments in our model to split people with a given feature vector σ over several bins. Our results also therefore apply to the natural special case of the model with *integral* risk assignments, in which all people with a given feature σ must go to the same bin.

Third, our model is a generalization of binary classification, which only allows for 2 bins. Note that although binary classification does not explicitly assign scores, we can consider the probability that an individual belongs to the positive class given that they were assigned to a specific bin to be the score for that bin. Thus, our results hold in the traditional binary classification setting as well.

Data-Generating Processes

Finally, there is the question of where the data in an instance of our problem comes from. Our results do not assume any particular process for generating the positive/negative class labels, feature vectors, and group memberships; we simply assume that we are given such a

collection of values (regardless of where they came from), and then our results address the existence or non-existence of certain risk assignments for these values.

This increases the generality of our results, since it means that they apply to any process that produces data of the form described by our model. To give an example of a natural generative model that would produce instances with the structure that we need, one could assume that each individual starts with a "hidden" class label (positive or negative), and a feature vector σ is then probabilistically generated for this individual from a distribution that can depend on their class label and their group membership. (If feature vectors produced for the two groups are disjoint from one another, then the requirement that the value of p_σ is independent of group membership given σ necessarily holds.) Since a process with this structure produces instances from our model, our results apply to data that arises from such a generative process.

1.3 Further Related Work

Mounting concern over discrimination in machine learning has led to a large body of new work seeking to better understand and prevent it. Barocas and Selbst survey a range of ways in which data-analysis algorithms can lead to discriminatory outcomes [2]. Kamiran and Calders, among others, seek to modify datasets to remove any information that might permit discrimination [13, 8]. Similarly, Zemel et al. look to learn fair intermediate representations of data while preserving information needed for classification [20].

One common notion of fairness, adopted by Feldman et al., Kamishima et al., and others, is "statistical parity" – equal fractions of each group should be treated as belonging to the positive class [8, 3, 13, 14]. Work in this direction has developed learning algorithms that penalize violations of statistical parity [3, 14]. As noted above, we consider definitions other than statistical parity that take into account the class membership (positive or negative) of the people being classified.

Dwork et al. propose a framework based on a task-specific externally defined similarity metric between individuals, seeking to achieve fairness through the goal that "similar people [be] treated similarly" [7]. They strive towards individual fairness, which is a stronger notion of fairness than the definitions we use; however, our approach shares some of the underlying motivation (though not the specifics) in that our balance conditions for the positive and negative classes also reflect the notion that similar people should be treated similarly.

Much of the applied work on risk scores, as noted above, focuses on calibration as a central goal [4, 6, 9]. In particular, responding to the criticism of their risk scores as displaying asymmetric errors for different groups, Dietrich et al. note that empirically, both in their domain and in similar settings, it is typically difficult to achieve symmetry in the error rates across groups when base rates differ significantly. Our formulation of the balance conditions for the positive and negative classes, and our result showing the incompability of these conditions with calibration, provides a theoretical basis for such observations.

2 The Characterization Theorems

Starting with the notation and definitions from the previous section, we now give a proof of Theorem 1.

First, let N_t be the number of people in group t. Since an $a_{t\sigma}$ fraction of the people in group t have feature vector σ, we write $n_{t\sigma} = a_{t\sigma} N_t$ for the number of people in group t with feature vector σ. Many of the components of the risk assignment and its evaluation

can be written in terms of operations on a set of underlying matrices and vectors, which we begin by specifying.

- First, let $|\sigma|$ denote the number of feature vectors in the instance, and let $p \in \mathbb{R}^{|\sigma|}$ be a vector indexed by the possible feature vectors, with the coordinate in position σ equal to p_σ. For group t, let $n_t \in \mathbb{R}^{|\sigma|}$ also be a vector indexed by the possible feature vectors, with the coordinate in position σ equal to $n_{t\sigma}$. Finally, it will be useful to have a representation of p as a diagonal matrix; thus, let P be a $|\sigma| \times |\sigma|$ diagonal matrix with $P_{\sigma\sigma} = p_\sigma$.
- We now specify a risk assignment as follows. The risk assignment involves a set of B bins with associated scores; let $v \in \mathbb{R}^B$ be a vector indexed by the bins, with the coordinate in position b equal to the score v_b of bin b. Let V be a diagonal matrix version of v: it is a $B \times B$ matrix with $V_{bb} = v_b$. Finally, let X be the $|\sigma| \times B$ matrix of $X_{\sigma b}$ values, specifying the fraction of people with feature vector σ who get mapped to bin b under the assignment procedure.

There is an important point to note about the $X_{\sigma b}$ values. If all of them are equal to 0 or 1, this corresponds to a procedure in which all people with the same feature vector σ get assigned to the same bin. When some of the $X_{\sigma b}$ values are not equal to 0 or 1, the people with vector σ are being divided among multiple bins. In this case, there is an implicit randomization taking place with respect to the positive and negative classes, and with respect to the two groups, which we can think of as follows. Since the procedure cannot distinguish among people with vector σ, in the case that it distributes these people across multiple bins, the subset of people with vector σ who belong to the positive and negative classes, and to the two groups, are divided up randomly across these bins in proportions corresponding to $X_{\sigma b}$. In particular, if there are $n_{t\sigma}$ group-t people with vector σ, the expected number of these people who belong to the positive class and are assigned to bin b is $n_{t\sigma} p_\sigma X_{\sigma b}$.

Let us now proceed with the proof of Theorem 1, starting with the assumption that our risk assignment satisfies conditions (A), (B), and (C).

Calibration within groups

We begin by working out some useful expressions in terms of the matrices and vectors defined above. We observe that $n_t^\top P$ is a vector in $\mathbb{R}^{|\sigma|}$ whose coordinate corresponding to feature vector σ equals the number of people in group t who have feature vector σ and belong to the positive class. $n_t^\top X$ is a vector in $\mathbb{R}^B$ whose coordinate corresponding to bin b equals the expected number of people in group t assigned to bin b.

By further multiplying these vectors on the right, we get additional useful quantities. Here are two in particular:

- $n_t^\top XV$ is a vector in $\mathbb{R}^B$ whose coordinate corresponding to bin b equals the expected sum of the scores assigned to all group-t people in bin b. That is, using the subscript b to denote the coordinate corresponding to bin b, we can write $(n_t^\top XV)_b = v_b(n_t^\top X)_b$ by the definition of the diagonal matrix V.
- $n_t^\top PX$ is a vector in $\mathbb{R}^B$ whose coordinate corresponding to bin b equals the expected number of group-t people in the positive class who are placed in bin b.

Now, condition (A), that the risk assignment is calibrated within groups, implies that the two vectors above are equal coordinate-wise, and so we have the following equation for all t:

$$n_t^\top PX = n_t^\top XV \tag{1}$$

Calibration condition (A) also has an implication for the total score received by all people in group t. Suppose we multiply the two sides of (1) on the right by the vector $\mathbf{e} \in \mathbb{R}^B$ whose coordinates are all 1, obtaining

$$n_t^\top PX\mathbf{e} = n_t^\top XV\mathbf{e}. \tag{2}$$

The left-hand-side is the number of group-t people in the positive class. The right-hand-side, which we can also write as $n_t^\top Xv$, is equal to the sum of the expected scores received by all group-t people. These two quantities are thus the same, and we write their common value as μ_t.

Fairness to the positive and negative classes

We now want to write down vector equations corresponding to the fairness conditions (B) and (C) for the negative and positive classes. First, recall that for the B-dimensional vector $n_t^\top PX$, the coordinate corresponding to bin b equals the expected number of group-t people in the positive class who are placed in bin b. Thus, to compute the sum of the expected scores received by all group-t people in the positive class, we simply need to take the inner product with the vector v, yielding $n_t^\top PXv$. Since μ_t is the total number of group-t people in the positive class, the average of the expected scores received by a group-t person in the positive class is the ratio $\frac{1}{\mu_t} n_t^\top PXv$. Thus, condition (C), that members of the positive class should receive the same average score in each group, can be written

$$\frac{1}{\mu_1} n_1^\top PXv = \frac{1}{\mu_2} n_2^\top PXv \tag{3}$$

Applying strictly analogous reasoning but to the fractions $1 - p_\sigma$ of people in the negative class, we we can write condition (B), that members of the negative class should receive the same average score in each group, as

$$\frac{1}{N_1 - \mu_1} n_1^\top (I - P)Xv = \frac{1}{N_2 - \mu_2} n_2^\top (I - P)Xv \tag{4}$$

Using (1), we can rewrite (3) to get

$$\frac{1}{\mu_1} n_1^\top XVv = \frac{1}{\mu_2} n_2^\top XVv \tag{5}$$

Similarly, we can rewrite (4) as

$$\frac{1}{N_1 - \mu_1} (\mu_1 - n_1^\top XVv) = \frac{1}{N_2 - \mu_2} (\mu_2 - n_2^\top XVv) \tag{6}$$

The portion of the score received by the positive class

We think of the ratios on the two sides of (3), and equivalently (5), as the average of the expected scores received by a member of the positive class in group t: the numerator is the sum of the expected scores received by the members of the positive class, and the denominator is the size of the positive class. Let us denote this fraction by γ_t. By (2), we can alternately think of the denominator as the sum of the expected scores received by all group-t people. Hence, the two sides of (3) and (5) can be viewed as representing the ratio of the sum of the expected scores in the positive class of group t to the sum of the expected scores in group t as a whole. (3) requires that $\gamma_1 = \gamma_2$; let us denote this common value by γ.

Now, we observe that $\gamma = 1$ corresponds to a case in which the sum of the expected scores in just the positive class of group t is equal to the sum of the expected scores in all of group t. In this case, it must be that all members of the negative class are assigned to bins of score 0. If any members of the positive class were assigned to a bin of score 0, this would violate the calibration condition (A); hence all members of the positive class are assigned to bins of positive score. Moreover, these bins of positive score contain no members of the negative class (since they've all been assigned to bins of score 0), and so again by the calibration condition (A), the members of the positive class are all assigned to bins of score 1. Finally, applying the calibration condition once more, it follows that the members of the negative class all have feature vectors σ with $p_\sigma = 0$ and the members of the positive class all have feature vectors σ with $p_\sigma = 1$. Hence, when $\gamma = 1$ we have perfect prediction.

Finally, we use our definition of γ_t as $\frac{1}{\mu_t} n_t^\top XVv$, and the fact that $\gamma_1 = \gamma_2 = \gamma$ to write (6) as

$$\begin{aligned}
\frac{1}{N_1 - \mu_1}(\mu_1 - \gamma\mu_1) &= \frac{1}{N_2 - \mu_2}(\mu_2 - \gamma\mu_2) \\
\frac{1}{N_1 - \mu_1}\mu_1(1-\gamma) &= \frac{1}{N_2 - \mu_2}\mu_2(1-\gamma) \\
\frac{\mu_1/N_1}{1 - \mu_1/N_1}(1-\gamma) &= \frac{\mu_2/N_2}{1 - \mu_2/N_2}(1-\gamma)
\end{aligned}$$

Now, this last equality implies that one of two things must be the case. Either $1 - \gamma = 0$, in which case $\gamma = 1$ and we have perfect prediction; or

$$\frac{\mu_1/N_1}{1 - \mu_1/N_1} = \frac{\mu_2/N_2}{1 - \mu_2/N_2},$$

in which case $\mu_1/N_1 = \mu_2/N_2$ and we have equal base rates. This completes the proof of Theorem 1.

Some Comments on the Connection to Statistical Parity

Earlier we noted that conditions (B) and (C) — the balance conditions for the positive and negative classes — are quite different from the requirement of *statistical parity*, which asserts that the average of the scores over *all* members of each group be the same.

When the two groups have equal base rates, then the risk assignment that gives the same score to everyone in the population achieves statistical parity along with conditions (A), (B), and (C). But when the two groups do not have equal base rates, it is immediate to show that statistical parity is inconsistent with both the calibration condition (A) and with the conjunction of the two balance conditions (B) and (C). To see the inconsistency of statistical parity with the calibration condition, we take Equation (1) from the proof above, sum the coordinates of the vectors on both sides, and divide by N_t, the number of people in group t. Statistical parity requires that the right-hand sides of the resulting equation be the same for $t = 1, 2$, while the assumption that the two groups have unequal base rates implies that the left-hand sides of the equation must be different for $t = 1, 2$. To see the inconsistency of statistical parity with the two balance conditions (B) and (C), we simply observe that if the average score assigned to the positive class and to the negative class are the same in the two groups, then the average of the scores over all members of the two groups cannot be the same provided they do not contain the same proportion of positive-class and negative-class members.

3 The Approximate Theorem

In this section we prove Theorem 2. First, we must first give a precise specification of the approximate fairness conditions:

$$(1-\varepsilon)[n_t^\top XV]_b \le [n_t^\top PX]_b \le (1-\varepsilon)[n_t^\top XV]_b \tag{A'}$$

$$\begin{aligned}(1-\varepsilon)\left(\frac{1}{N_2-\mu_2}\right) n_2^\top (I-P)Xv &\le \left(\frac{1}{N_1-\mu_1}\right) n_1^\top (I-P)Xv \\ &\le (1+\varepsilon)\left(\frac{1}{N_2-\mu_2}\right) n_2^\top (I-P)Xv \end{aligned} \tag{B'}$$

$$(1-\varepsilon)\left(\frac{1}{\mu_2}\right) n_2^\top PXv \le \left(\frac{1}{\mu_1}\right) n_1^\top PXv \le (1+\varepsilon)\left(\frac{1}{\mu_2}\right) n_2^\top PXv \tag{C'}$$

For (B') and (C'), we also require that these hold when μ_1 and μ_2 are interchanged.

We also specify the approximate versions of perfect prediction and equal base rates in terms of $f(\varepsilon)$, which is a function that goes to 0 as ε goes to 0.

- *Approximate perfect prediction.* $\gamma_1 \ge 1 - f(\varepsilon)$ and $\gamma_2 \ge 1 - f(\varepsilon)$
- *Approximately equal base rates.* $|\mu_1/N_1 - \mu_2/N_2| \le f(\varepsilon)$

A brief overview of the proof of Theorem 2 is as follows. It proceeds by first establishing an approximate form of Equation (1) above, which implies that the total expected score assigned in each group is approximately equal to the total size of the positive class. This in turn makes it possible to formulate approximate forms of Equations (3) and (4). When the base rates are close together, the approximation is too loose to derive bounds on the predictive power; but this is okay since in this case we have approximately equal base rates. Otherwise, when the base rates differ significantly, we show that most of the expected score must be assigned to the positive class, giving us approximately perfect prediction.

The remainder of this section provides the full details of the proof.

Total scores and the number of people in the positive class

First, we will show that the total score for each group is approximately μ_t, the number of people in the positive class. Define $\hat{\mu}_t = n_t^\top Xv$. Using (A'), we have

$$\begin{aligned}\hat{\mu}_t &= n_t^\top Xv \\ &= n_t^\top XVe \\ &= \sum_{b=1}^{B} [n_t^\top PX]_b \\ &\le (1+\varepsilon)\sum_{b=1}^{B} [n_t^\top PX]_b \\ &= (1+\varepsilon) n_t^\top PXe \\ &= (1+\varepsilon)\mu_t\end{aligned}$$

Similarly, we can lower bound $\hat{\mu}_t$ as

$$\begin{aligned}\hat{\mu}_t &= \sum_{b=1}^{B} [n_t^\top PX]_b \\ &\geq (1-\varepsilon) \sum_{b=1}^{B} [n_t^\top PX]_b \\ &= (1-\varepsilon)\mu_t\end{aligned}$$

Combining these, we have

$$(1-\varepsilon)\mu_t \leq \hat{\mu}_t \leq (1+\varepsilon)\mu_t. \tag{7}$$

The portion of the score received by the positive class

We can use (C') to show that $\gamma_1 \approx \gamma_2$. Recall that γ_t, the average of the expected scores assigned to members of the positive class in group t, is defined as $\gamma_t = \frac{1}{\mu_t} n_t PXv$. Then, it follows trivially from (C') that

$$(1-\varepsilon)\gamma_2 \leq \gamma_1 \leq (1+\varepsilon)\gamma_2. \tag{8}$$

The relationship between the base rates

We can apply this to (B') to relate μ_1 and μ_2, using the observation that the score not received by people of the positive class must fall instead to people of the negative class. Examining the left inequality of (B'), we have

$$\begin{aligned}(1-\varepsilon)\left(\frac{1}{N_2-\mu_2}\right) n_t^\top (I-P)Xv &= (1-\varepsilon)\left(\frac{1}{N_2-\mu_2}\right)(n_t^\top Xv - n_t^\top PXv) \\ &= (1-\varepsilon)\left(\frac{1}{N_2-\mu_2}\right)(\hat{\mu}_2 - \gamma_2\mu_2) \\ &\geq (1-\varepsilon)\left(\frac{1}{N_2-\mu_2}\right)((1-\varepsilon)\mu_2 - \gamma_2\mu_t) \\ &= (1-\varepsilon)\left(\frac{\mu_2}{N_2-\mu_2}\right)(1-\varepsilon-\gamma_2) \\ &\geq (1-\varepsilon)\left(\frac{\mu_2}{N_2-\mu_2}\right)\left(1-\varepsilon-\frac{\gamma_1}{1-\varepsilon}\right) \\ &= (1-2\varepsilon+\varepsilon^2-\gamma_1)\left(\frac{\mu_2}{N_2-\mu_2}\right)\end{aligned}$$

Thus, the left inequality of (B') becomes

$$(1-2\varepsilon+\varepsilon^2-\gamma_1)\left(\frac{\mu_2}{N_2-\mu_2}\right) \leq \left(\frac{1}{N_1-\mu_1}\right) n_t^\top (I-P)Xv \tag{9}$$

By definition, $\hat{\mu}_1 = n_t^\top Xv$ and $\gamma_t\mu_t = n_t^\top PXv$, so this becomes

$$(1-2\varepsilon+\varepsilon^2-\gamma_1)\left(\frac{\mu_2}{N_2-\mu_2}\right) \leq \left(\frac{1}{N_1-\mu_1}\right)(\hat{\mu}_1 - \gamma_1\mu_1) \tag{10}$$

If the base rates differ

Let ρ_1 and ρ_2 be the respective base rates, i.e. $\rho_1 = \mu_1/N_1$ and $\rho_2 = \mu_2/N_2$. Assume that $\rho_1 \leq \rho_2$ (otherwise we can switch μ_1 and μ_2 in the above analysis), and assume towards contradiction that the base rates differ by at least $\sqrt{\varepsilon}$, meaning $\rho_1 + \sqrt{\varepsilon} < \rho_2$. Using (10),

$$\frac{\rho_1 + \sqrt{\varepsilon}}{1 - \rho_1 - \sqrt{\varepsilon}} \leq \frac{\rho_2}{1 - \rho_2} \leq \left(\frac{1 + \varepsilon - \gamma_1}{1 - 2\varepsilon + \varepsilon^2 - \gamma_1}\right)\left(\frac{\rho_1}{1 - \rho_1}\right)$$

Simplifiying,

$$\begin{aligned}
(\rho_1 + \sqrt{\varepsilon})(1 - \rho_1)(1 - 2\varepsilon + \varepsilon^2 - \gamma_1) &\leq \rho_1(1 - \rho_1 - \sqrt{\varepsilon})(1 + \varepsilon - \gamma_1) \\
(\rho_1 + \sqrt{\varepsilon})(1 - \rho_1)(1 - 2\varepsilon) - \rho_1(1 - \rho_1 - \sqrt{\varepsilon})(1 + \varepsilon) &\leq \gamma_1[(\rho_1 + \sqrt{\varepsilon})(1 - \rho_1) \\
&\quad - \rho_1(1 - \rho_1 - \sqrt{\varepsilon})] \\
(\rho_1 + \sqrt{\varepsilon})(1 - \rho_1)(1 - 2\varepsilon) - \rho_1(1 - \rho_1 - \sqrt{\varepsilon})(1 + \varepsilon) &\leq \gamma_1[\sqrt{\varepsilon}(1 - \rho_1) + \sqrt{\varepsilon}\rho_1]
\end{aligned}$$

$$\begin{aligned}
\rho_1[(1 - \rho_1)(1 - 2\varepsilon) - (1 - \rho_1 - \sqrt{\varepsilon})(1 + \varepsilon)] + \sqrt{\varepsilon}(1 - \rho_1)(1 - 2\varepsilon) &\leq \gamma_1\sqrt{\varepsilon} \\
\rho_1(-2\varepsilon + 2\varepsilon\rho_1 - \varepsilon + \varepsilon\rho_1 + \sqrt{\varepsilon} + \varepsilon\sqrt{\varepsilon}) + \sqrt{\varepsilon}(1 - 2\varepsilon - \rho_1 + 2\varepsilon\rho_1) &\leq \gamma_1\sqrt{\varepsilon} \\
\rho_1(-3\varepsilon + 3\varepsilon\rho_1 + \sqrt{\varepsilon} + \varepsilon\sqrt{\varepsilon} - \sqrt{\varepsilon} + 2\varepsilon\sqrt{\varepsilon}) + \sqrt{\varepsilon}(1 - 2\varepsilon) &\leq \gamma_1\sqrt{\varepsilon} \\
\varepsilon\rho_1(-3 + 3\rho_1 + 3\sqrt{\varepsilon}) + \sqrt{\varepsilon}(1 - 2\varepsilon) &\leq \gamma_1\sqrt{\varepsilon} \\
3\varepsilon\rho_1(-1 + \rho_1) + \sqrt{\varepsilon}(1 - 2\varepsilon) &\leq \gamma_1\sqrt{\varepsilon} \\
1 - 2\varepsilon - 3\sqrt{\varepsilon}\rho_1(1 - \rho_1) &\leq \gamma_1 \\
1 - \sqrt{\varepsilon}\left(2\sqrt{\varepsilon} + \frac{3}{4}\right) &\leq \gamma_1
\end{aligned}$$

Recall that $\gamma_2 \geq \gamma_1(1 - \varepsilon)$, so

$$\begin{aligned}
\gamma_2 &\geq (1 - \varepsilon)\gamma_1 \\
&\geq (1 - \varepsilon)\left(1 - \sqrt{\varepsilon}\left(2\sqrt{\varepsilon} + \frac{3}{4}\right)\right) \\
&\geq 1 - \varepsilon - \sqrt{\varepsilon}\left(2\sqrt{\varepsilon} + \frac{3}{4}\right) \\
&= 1 - \sqrt{\varepsilon}\left(3\sqrt{\varepsilon} + \frac{3}{4}\right)
\end{aligned}$$

Let $f(\varepsilon) = \sqrt{\varepsilon}\max(1, 3\sqrt{\varepsilon} + 3/4)$. Note that we assumed that ρ_1 and ρ_2 differ by an additive $\sqrt{\varepsilon} \leq f(\varepsilon)$. Therefore if the ε-fairness conditions are met and the base rates are not within an additive $f(\varepsilon)$, then $\gamma_1 \geq 1 - f(\varepsilon)$ and $\gamma_2 \geq 1 - f(\varepsilon)$. This completes the proof of Theorem 2.

4 Reducing Loss with Equal Base Rates

In a risk assigment, we would like as much of the score as possible to be assigned to members of the positive class. With this in mind, if an individual receives a score of v, we define their *individual loss* to be v if they belong to the negative class, and $1 - v$ if they belong to the positive class. The loss of the risk assignment in group t is then the sum of the expected individual losses to each member of group t. In terms of the matrix-vector products used in

the proof of Theorem 1, one can show that the loss for group t may be written as

$$\begin{aligned}\ell_t(X) &= n_t^\top (I - P)Xv + (\mu_t - n_t^\top PXv) \\ &= 2(\mu_t - n_t^\top PXv),\end{aligned}$$

and the total loss is just the weighted sum of the losses for each group.

Now, let us say that a *fair assignment* is one that satisfies our three conditions (A), (B), and (C). As noted above, when the base rates in the two groups are equal, the set of fair assignments is non-empty, since the calibrated risk assignment that places everyone in a single bin is fair. We can therefore ask, in the case of equal base rates, whether there exists a fair assignment whose loss is strictly less than that of the trivial one-bin assignment. It is not hard to show that this is possible if and only if there is any assignment using more than one bin; we will call such an assignment a *non-trivial assignment*.

Note that the assignment that minimizes loss is simply the one that assigns each σ to a separate bin with a score of p_σ, meaning X is the identity matrix. While this assignment, which we refer to as the identity assignment I, is well-calibrated, it may violate fairness conditions (B) and (C). It is not hard to show that the loss for any other assignment is strictly greater than the loss for I. As a result, unless the identity assignment happens to be fair, every fair assignment must have larger loss than that of I, forcing a tradeoff between performance and fairness.

4.1 Characterization of Well-Calibrated Solutions

To better understand the space of feasible solutions, suppose we drop the fairness conditions (B) and (C) for now and study risk assignments that are simply well-calibrated, satisfying (A). As in the proof of Theorem 1, we write γ_t for the average of the expected scores assigned to members of the positive class in group t, and we define the *fairness difference* to be $\gamma_1 - \gamma_2$. If this is nonnegative, we say the risk assignment *weakly favors* group 1; if it is nonpositive, it weakly favors group 2. Since a risk assignment is fair if and only if $\gamma_1 = \gamma_2$, it is fair if and only if the fairness difference is 0.

We wish to characterize when non-trivial fair risk assignments are possible. First, we observe that without the fairness requirements, the set of possible fairness differences under well-calibrated assignments is an interval.

▶ **Lemma 3.** *If group 1 and group 2 have equal base rates, then for any two non-trivial well-calibrated risk assignments with fairness differences d_1 and d_2 and for any $d_3 \in [d_1, d_2]$, there exists a non-trivial well-calibrated risk assignment with fairness difference d_3.*

Proof. The basically idea is that we can effectively take convex combinations of well-calibrated assignments to produce any well-calibrated assignment "in between" them. We carry this out as follows.

Let $X^{(1)}$ and $X^{(2)}$ be the allocation matrices for assignments with fairness differences d_1 and d_2 respectively, where $d_1 < d_2$. Choose λ such that $\lambda d_1 + (1 - \lambda)d_2 = d_3$, meaning $\lambda = (d_2 - d_3)/(d_2 - d_1)$. Then, $X^{(3)} = [\lambda X^{(1)} \quad (1 - \lambda)X^{(2)}]$ is a nontrivial well-calibrated assignment with fairness difference d_3.

First, we observe that $X^{(3)}$ is a valid assignment because each row sums to 1 (meaning everyone from every σ gets assigned to a bin), since each row of $\lambda X^{(1)}$ sums to λ and each row of $(1 - \lambda)X^{(2)}$ sums to $(1 - \lambda)$. Moreover, it is nontrivial because every nonempty bin created by $X^{(1)}$ and $X^{(2)}$ is a nonempty bin under $X^{(3)}$.

Let $v^{(1)}$ and $v^{(2)}$ be the respective bin labels for assignments $X^{(1)}$ and $X^{(2)}$. Define $v^{(3)} = \begin{bmatrix} v^{(1)} \\ v^{(2)} \end{bmatrix}$.

Finally, let $V^{(3)} = \text{diag}(v^{(3)})$. Define $V^{(1)}$ and $V^{(2)}$ analogously. Note that

$$V^{(3)} = \begin{bmatrix} V^{(1)} & 0 \\ 0 & V^{(2)} \end{bmatrix}.$$

We observe that $X^{(3)}$ is calibrated because

$$\begin{aligned} n_t^\top P X^{(3)} &= n_t^\top P[\lambda X^{(1)} \quad (1-\lambda)X^{(2)}] \\ &= [\lambda n_t^\top P X^{(1)} \quad (1-\lambda) n_t^\top P X^{(2)}] \\ &= [\lambda n_t^\top X^{(1)} V^{(1)} \quad (1-\lambda) n_t^\top X^{(2)} V^{(2)}] \\ &= n_t^\top [\lambda X^{(1)} \quad (1-\lambda)X^{(2)}] V^{(3)} \\ &= n_t^\top X^{(3)} V^{(3)} \end{aligned}$$

Finally, we show that the fairness difference is d_3. Let $\gamma_1^{(1)}$ and $\gamma_2^{(1)}$ be the portions of the total expected score receieved by the positive class from each group respectively. Define $\gamma_1^{(2)}, \gamma_2^{(2)}, \gamma_1^{(3)}, \gamma_2^{(3)}$ similarly.

$$\begin{aligned} \gamma_1^{(3)} - \gamma_2^{(3)} &= \frac{1}{\mu} n_1^\top P X^{(3)} v^{(3)} - \frac{1}{\mu} n_2^\top P X^{(3)} v^{(3)} \\ &= \frac{1}{\mu}(n_1^\top - n_2^\top) P X^{(3)} v^{(3)} \\ &= \frac{1}{\mu}(n_1^\top - n_2^\top) P[\lambda X^{(1)} v^{(1)} \quad (1-\lambda) X^{(2)} v^{(2)}] \\ &= \frac{1}{\mu}(\lambda(n_1^\top - n_2^\top) P X^{(1)} v^{(1)} + (1-\lambda)(n_1^\top - n_2^\top) X^{(2)} v^{(2)}]) \\ &= \lambda(\gamma_1^{(1)} - \gamma_2^{(1)}) + (1-\lambda)(\gamma_1^{(2)} - \gamma_2^{(2)}) \\ &= \lambda d_1 + (1-\lambda) d_2 \\ &= d_3 \end{aligned}$$

◀

▶ **Corollary 4.** *There exists a non-trivial fair assignment if and only if there exist non-trivial well-calibrated assignments $X^{(1)}$ and $X^{(2)}$ such that $X^{(1)}$ weakly favors group 1 and $X^{(2)}$ weakly favors group 2.*

Proof. If there is a non-trivial fair assignment, then it weakly favors both group 1 and group 2, proving one direction.

To prove the other direction, observe that the fairness differences d_1 and d_2 of $X^{(1)}$ and $X^{(2)}$ are nonnegative and nonpositive respectively. Since the set of fairness differences achievable by non-trivial well-calibrated assignments is an interval by Lemma 3, there exists a non-trivial well-calibrated assignment with fairness difference 0, meaning there exists a non-trivial fair assignment. ◀

It is an open question whether there is a polynomial-time algorithm to find a fair assignment of minimum loss, or even to determine whether a non-trivial fair solution exists.

4.2 NP-Completeness of Non-Trivial Integral Fair Risk Assignments

As discussed in the introduction, risk assignments in our model are allowed to split people with a given feature vector σ over several bins; however, it is also of interest to consider the special case of *integral* risk assignments, in which all people with a given feature σ must go to the same bin. For the case of equal base rates, we can show that determining whether there is a non-trivial integral fair assignment is NP-complete. The proof uses a reduction from the Subset Sum problem and is given in the Appendix.

The basic idea of the reduction is as follows. We have an instance of Subset Sum with numbers $x_1, \ldots, x_n$ and a target number T; the question is whether there is a subset of the x_i's that sums to T. As before, γ_t denotes the average of the expected scores received by members of the positive class in group t. We first ensure that there is exactly one non-trivial way to allocate the people of group 1, allowing us to control γ_1. The fairness conditions then require that $\gamma_2 = \gamma_1$, which we can use to encode the target value in the instance of Subset Sum. For every input number x_i in the Subset Sum instance, we create $p_{\sigma_{2i-1}}$ and $p_{\sigma_{2i}}$, close to each other in value and far from all other p_σ values, such that grouping σ_{2i-1} and σ_{2i} together into a bin corresponds to choosing x_i for the subset, while not grouping them corresponds to not taking x_i. This ensures that group 2 can be assigned with the correct value of γ_2 if and only if there is a solution to the Subset Sum instance.

5 Conclusion

In this work we have formalized three fundamental conditions for risk assignments to individuals, each of which has been proposed as a basic measure of what it means for the risk assignment to be fair. Our main results show that except in highly constrained special cases, it is not possible to satisfy these three constraints simultaneously; and moreover, a version of this fact holds in an approximate sense as well.

Since these results hold regardless of the method used to compute the risk assignment, it can be phrased in fairly clean terms in a number of domains where the trade-offs among these conditions do not appear to be well-understood. To take one simple example, suppose we want to determine the risk that a person is a carrier for a disease X, and suppose that a higher fraction of women than men are carriers. Then our results imply that in any test designed to estimate the probability that someone is a carrier of X, at least one of the following undesirable properties must hold: (a) the test's probability estimates are systematically skewed upward or downward for at least one gender; or (b) the test assigns a higher average risk estimate to healthy people (non-carriers) in one gender than the other; or (c) the test assigns a higher average risk estimate to carriers of the disease in one gender than the other. The point is that this trade-off among (a), (b), and (c) is not a fact about medicine; it is simply a fact about risk estimates when the base rates differ between two groups.

It is also interesting to note that the basic set-up of our model, with the population divided across a set of feature vectors that convey no information about race, is in fact a very close match to the information one gets from the output of a well-calibrated risk tool. In this sense, one setting for our model would be the problem of applying post-processing to the output of such a risk tool to ensure additional fairness guarantees. Indeed, since much of the recent controversy about fair risk scores has involved risk tools that are well-calibrated but lack the other fairness conditions we consider, such an interpretation of the model could be a useful way to think about how one might work with these tools in the context of a broader system.

Finally, we note that our results suggest a number of interesting directions for further work. First, when the base rates between the two underlying groups are equal, our results

do not resolve the computational tractability of finding the most accurate risk assignment, subject to our three fairness conditions, when the people with a given feature vector can be split across multiple bins. (Our NP-completeness result applies only to the case in which everyone with a given feature vector must be assigned to the same bin.) Second, there may be a number of settings in which the cost (social or otherwise) of false positives may differ greatly from the cost of false negatives. In such cases, we could imagine searching for risk assignments that satisfy the calibration condition together with only one of the two balance conditions, corresponding to the class for whom errors are more costly. Determining when two of our three conditions can be simultaneously satisfied in this way is an interesting open question. More broadly, determining how the trade-offs discussed here can be incorporated into broader families of proposed fairness conditions suggests interesting avenues for future research.

References

1 Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals. And it's biased against blacks. *ProPublica*, May 23, 2016. URL: `https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing`.

2 Solon Barocas and Andrew D Selbst. Big data's disparate impact. *California Law Review*, 104, 2016.

3 Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data Mining and Knowledge Discovery*, 21(2):277–292, 2010.

4 Cynthia S. Crowson, Elizabeth J. Atkinson, and Terry M. Therneau. Assessing calibration of prognostic risk scores. *Statistical Methods in Medical Research*, 25(4):1692–1706, 2016.

5 Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings. *Proceedings on Privacy Enhancing Technologies*, 2015(1):92–112, 2015. URL: `http://www.degruyter.com/view/j/popets.2015.1.issue-1/popets-2015-0007/popets-2015-0007.xml`.

6 William Dieterich, Christina Mendoza, and Tim Brennan. COMPAS risk scales: Demonstrating accuracy equity and predictive parity. Technical report, Northpointe, July 2016. URL: `http://www.northpointeinc.com/northpointe-analysis`.

7 Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 214–226, 2012. `doi:10.1145/2090236.2090255`.

8 Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 259–268, New York, NY, USA, 2015. ACM. `doi:10.1145/2783258.2783311`.

9 Anthony Flores, Christopher Lowenkamp, and Kristin Bechtel. False positives, false negatives, and false analyses: A rejoinder to "machine bias: There's software used across the country to predict future criminals. and it's biased against blacks.". Technical report, Crime & Justice Institute, September 2016. URL: `http://www.crj.org/cji/entry/false-positives-false-negatives-and-false-analyses-a-rejoinder`.

10 Dean P. Foster and Rakesh V. Vohra. Asymptotic calibration. *Biometrika*, 85(2):379–390, 1998.

11 Howard N. Garb. Race bias, social class bias, and gender bias in clinical judgment. *Clinical Psychology: Science and Practice*, 4(2):99–120, 1997.

12 Abe Gong. Ethics for powerful algorithms (1 of 4). *Medium*, July 12, 2016. URL: https://medium.com/@AbeGong/ethics-for-powerful-algorithms-1-of-3-a060054efd84#.dhsd2ut3i.

13 Faisal Kamiran and Toon Calders. Classifying without discriminating. *2009 2nd International Conference on Computer, Control and Communication, IC4 2009*, 2009. doi:10.1109/IC4.2009.4909197.

14 Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. Fairness-aware learning through regularization approach. *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 643–650, 2011. doi:10.1109/ICDMW.2011.83.

15 Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the COMPAS recidivism algorithm. *ProPublica*, May 23, 2016. URL: https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm.

16 Executive Office of the President. Big data: A report on algorithmic systems, opportunity, and civil rights. Technical report, The White House, Washington, USA, May 2016.

17 Propublica analysis. URL: https://docs.google.com/document/d/1pKtyl8XmJH7Z09lxkb70n6fa2Fiitd7ydbxgCT_wCXs/edit?pref=2&pli=1.

18 Latanya Sweeney. Discrimination in online ad delivery. *Communications of the ACM*, 56(5):44–54, 2013. doi:10.1145/2447976.2447990.

19 David R. Williams and Selina A. Mohammed. Discrimination and racial disparities in health: Evidence and needed research. *J. Med. Behav.*, 32(1), 2009.

20 Richard S Zemel, Yu Wu, Kevin Swersky, Toniann Pitassi, and Cynthia Dwork. Learning Fair Representations. *Proceedings of the 30th International Conference on Machine Learning*, 28:325–333, 2013. URL: http://jmlr.org/proceedings/papers/v28/zemel13.html.

A Appendix: NP-Completeness of Non-Trivial Integral Fair Risk Assignments

We can reduce to the integral assignment problem, parameterized by $a_{1\sigma}, a_{2\sigma}$, and p_σ, from subset sum as follows.

Given n numbers $x_1, \ldots, x_n$ and a target T, we create an instance of the integral assignment problem with $\sigma_1, \ldots, \sigma_{2n+2}$. $a_{1,\sigma_i} = 1/2$ if $i \in \{2n+1, 2n+2\}$ and 0 otherwise. $a_{2,\sigma_i} = 1/(2n)$ if $i \leq 2n$ and 0 otherwise. We make the following definitions:

$$\begin{aligned}
\hat{x}_i &= x_i/(Tn^4) \\
\varepsilon_i &= \sqrt{\hat{x}_i/2} \\
p_{\sigma_{2i-1}} &= i/(n+1) - \varepsilon_i && (1 \leq i \leq n) \\
p_{\sigma_{2i}} &= i/(n+1) + \varepsilon_i && (1 \leq i \leq n) \\
\gamma &= 1/n \sum_{i=1}^{2n} p_{\sigma_i}^2 - 1/n^5 \\
p_{\sigma_{2n+1}} &= (1 - \sqrt{2\gamma - 1})/2 \\
p_{\sigma_{2n+2}} &= (1 + \sqrt{2\gamma - 1})/2
\end{aligned}$$

With this definition, the subset sum instance has a solution if and only if the integral assignment instance given by $a_{1,\sigma}, a_{2,\sigma}, p_{\sigma_1}, \ldots, p_{\sigma_{2n+2}}$ has a solution.

Before we prove this, we need the following lemma.

▶ **Lemma 5.** *For any* $y_1, \ldots, y_k \in \mathbb{R}$,

$$\sum_{i=1}^{k} y_i^2 - \frac{1}{k}\left(\sum_{i=1}^{n} y_i\right)^2 = \frac{1}{k}\sum_{i<j}^{k}(y_i - y_j)^2$$

Proof.

$$\begin{aligned}
\sum_{i=1}^{k} y_i^2 - \frac{1}{k}\left(\sum_{i=1}^{n} y_i\right)^2 &= \sum_{i=1}^{k} y_i^2 - \frac{1}{k}\left(\sum_{i=1}^{k} y_i^2 + 2\sum_{i<j}^{k} y_i y_j\right) \\
&= \frac{k-1}{k}\sum_{i=1}^{k} y_i^2 - \frac{2}{k}\sum_{i<j}^{k} y_i y_j \\
&= \frac{1}{k}\sum_{i<j}^{k}(y_i^2 + y_j^2) - \frac{2}{k}\sum_{i<j}^{k} y_i y_j \\
&= \frac{1}{k}\sum_{i<j}^{k} y_i^2 - 2y_i y_j + y_j^2 \\
&= \frac{1}{k}\sum_{i<j}^{k}(y_i - y_j)^2
\end{aligned}$$

◀

Now, we can prove that the integral assignment problem is NP-hard.

Proof. First, we observe that for any nontrivial solution to the integral assignment instance, there must be two bins $b \neq b'$ such that $X_{\sigma_{2n+1},b} = 1$ and $X_{\sigma_{2n+2},b'} = 1$. In other words, the people with σ_{2n+1} and σ_{2n+2} must be split up. If not, then all the people of group 1 would be in the same bin, meaning that bin must be labeled with the base rate $\rho_1 = 1/2$. In order to maintain fairness, the same would have to be done for all the people of group 2, resulting in the trivial solution. Moreover, b and b' must be labeled $(1 \pm \sqrt{2\gamma - 1})/2$ respectively because those are the fraction of people of group 1 in those bins who belong to the positive class.

This means that $\gamma_1 = 1/\rho \cdot (a_{1,\sigma_{2n+1}} p_{\sigma_{2n+1}}^2 + a_{1,\sigma_{2n+2}} p_{\sigma_{2n+2}}^2) = p_{\sigma_{2n+1}}^2 + p_{\sigma_{2n+2}}^2 = \gamma$ as defined above. We know that a well-calibrated assignment is fair if and only if $\gamma_1 = \gamma_2$, so we know $\gamma_2 = \gamma$.

Next, we observe that $\rho_2 = \rho_1 = 1/2$ because all of the positive $a_{2,\sigma}$'s are $1/2n$, so ρ_2 is just the average of $\{p_{\sigma_1}, \ldots, p_{\sigma_{2n}}\}$, which is $1/2$ by symmetry.

Let Q be the partition of $[2n]$ corresponding to the assignment, meaning that for a given $q \in Q$, there is a bin b_q containing all people with σ_i such that $i \in q$. The label on that bin is

$$\begin{aligned}
v_q &= \frac{\sum_{i \in q} a_{2,\sigma_i} p_{\sigma_i}}{\sum_{i \in q} a_{2,\sigma_i}} \\
&= \frac{1/2n \sum_{i \in q} p_{\sigma_i}}{|q|/2n} \\
&= \frac{1}{|q|}\sum_{i \in q} p_{\sigma_i}
\end{aligned}$$

Furthermore, bin b_q contains $\sum_{i \in q} a_{2,\sigma_i} p_{\sigma_i} = 1/2n \sum_{i \in q} p_{\sigma_i}$ positive fraction. Using this, we can come up with an expression for γ_2.

$$\begin{aligned}\gamma_2 &= \frac{1}{\rho} \sum_{q \in Q} \left(v_b \cdot \frac{1}{2n} \sum_{i \in q} p_{\sigma_i} \right) \\ &= \frac{1}{n} \sum_{q \in Q} \frac{1}{|q|} \left(\sum_{i \in q} p_{\sigma_i} \right)^2\end{aligned}$$

Setting this equal to γ, we have

$$\begin{aligned}\frac{1}{n} \sum_{q \in Q} \frac{1}{|q|} \left(\sum_{i \in q} p_{\sigma_i} \right)^2 &= \frac{1}{n} \sum_{i=1}^{2n} p_{\sigma_i}^2 - \frac{1}{n^5} \\ \sum_{q \in Q} \frac{1}{|q|} \left(\sum_{i \in q} p_{\sigma_i} \right)^2 &= \sum_{i=1}^{2n} p_{\sigma_i}^2 - \frac{1}{n^4}\end{aligned}$$

Subtracting both sides from $\sum_{i=1}^{2n} p_{\sigma_i}^2$ and using Lemma 5, we have

$$\sum_{q \in Q} \frac{1}{|q|} \sum_{i<j \in q} (p_{\sigma_i} - p_{\sigma_j})^2 = \frac{1}{n^4} \tag{11}$$

Thus, Q is a fair nontrivial assignment if and only if (11) holds.

Next, we show that there exists Q that satisfies (11) if and only if there there exists some $S \subseteq [n]$ such that $\sum_{i \in S} \hat{a}_i = 1/n^4$.

Assume Q satisfies (11). Then, we first observe that any $q \in Q$ must either contain a single i, meaning it does not contribute to the left hand side of (11), or $q = \{2i-1, 2i\}$ for some i. To show this, observe that the closest two elements of $\{p_{\sigma_1}, \ldots, p_{\sigma_{2n}}\}$ not of the form $\{p_{\sigma_{2i-1}}, p_{\sigma_{2i}}\}$ must be some $\{p_{\sigma_{2i}}, p_{\sigma_{2i+1}}\}$. However, we find that

$$
\begin{aligned}
(p_{\sigma_{2i+1}} - p_{\sigma_{2i}})^2 &= \left(\frac{i+1}{n+1} - \varepsilon_{i+1} - \left(\frac{i}{n+1} + \varepsilon_i\right)\right)^2 \\
&= \left(\frac{1}{n+1} - \varepsilon_{i+1} - \varepsilon_i\right)^2 \\
&= \left(\frac{1}{n+1} - \sqrt{\frac{\hat{x}_{i+1}}{2}} - \sqrt{\frac{\hat{x}_i}{2}}\right)^2 \\
&\geq \left(\frac{1}{n+1} - \sqrt{\frac{2}{n^4}}\right)^2 && (\hat{x}_i \leq 1/n^4) \\
&= \left(\frac{1}{n+1} - \frac{\sqrt{2}}{n^2}\right)^2 \\
&\geq \left(\frac{1}{2n} - \frac{\sqrt{2}}{n^2}\right)^2 \\
&= \left(\frac{n - 2\sqrt{2}}{2n^2}\right)^2 \\
&\geq \left(\frac{n}{4n^2}\right)^2 \\
&= \left(\frac{1}{4n}\right)^2 \\
&= \frac{1}{16n^2}
\end{aligned}
$$

If any q contains any j, k not of the form $2i-1, 2i$, then (11) will have a term on the left hand side at least $1/n \cdot 1/(16n^2) = 1/(16n^3) > 1/n^4$ for large enough n, and since there can be no negative terms on the left hand side, this immediately makes it impossible for Q to satisify (11).

Consider every $2i-1, 2i \in [2n]$. Let $q_i = \{2i-1, 2i\}$. As shown above, either $q_i \in Q$ or $\{2i-1\} \in Q$ and $\{2i\} \in Q$. In the latter case, neither $p_{\sigma_{2i-1}}$ nor $p_{\sigma_{2i}}$ contributes to (11). If $q_i \in Q$, then q_i contributes $1/2(p_{\sigma_{2i} - p_{\sigma_{2i-1}}})^2 = 1/2(2\varepsilon_i)^2 = \hat{x}_i$ to the overall sum on the left hand side. Therefore, we can write the left hand side of (11) as

$$\sum_{q \in Q} \frac{1}{|q|} \sum_{i<j \in q} (p_{\sigma_i} - p_{\sigma_j})^2 = \sum_{q_i \in Q} \frac{1}{2} (p_{\sigma_{2i} - p_{\sigma_{2i-1}}})^2 = \sum_{q_i \in Q} \hat{x}_i = \frac{1}{n^4}$$

Then, we can build a solution to the original subset sum instance as $S = \{i : q_i \in Q\}$, giving us $\sum_{i \in S} \hat{x}_i = \frac{1}{n^4}$. Multiplying both sides by Tn^4, we get $\sum_{i \in S} x_i = T$, meaning S is a solution for the subset sum instance.

To prove the other direction, assume we have a solution $S \subseteq [n]$ such that $\sum_{i \in S} x_i = T$. Dividing both sides by Tn^4, we get $\sum_{i \in S} \hat{x}_i = 1/n^4$. We build a partition Q of $2n$ by starting with the empty set and adding $q_i = \{2i-1, 2i\}$ to Q if $i \in S$ and $\{2i-1\}$ and $\{2i\}$ to Q otherwise. Clearly, each element of $[2n]$ appears in Q at most once, making this a valid partition. Moreover, when checking to see if (11) is satisfied (which is true if and only if Q is a fair assignment), we can ignore all $q \in Q$ such that $|q| = 1$ because they don't contribute to the left hand side. Since, we again have

$$\sum_{q \in Q} \frac{1}{|q|} \sum_{i<j \in q} (p_{\sigma_i} - p_{\sigma_j})^2 = \sum_{q_i \in Q} \frac{1}{2} (p_{\sigma_{2i} - p_{\sigma_{2i-1}}})^2 = \sum_{q_i \in Q} \hat{x}_i = \frac{1}{n^4}$$

meaning Q is a fair assignment. This completes the reduction. ◀

We have shown that the integral assignment problem is NP-hard, and it is clearly in NP because given an integral assignment, we can verify in polynomial time whether such an assignment satisfies the conditions (A), (B), and (C). Thus, the integral assignment problem is NP-complete.

Non-Backtracking Spectrum of Degree-Corrected Stochastic Block Models

Lennart Gulikers[1], Marc Lelarge[2], and Laurent Massoulié[3]

1 **Microsoft Research - INRIA Joint Centre, Palaiseau and École Normale Supérieure, Paris, France**
lennart.gulikers@inria.fr
2 **INRIA Paris and École Normale Supérieure, Paris, France**
marc.lelarge@ens.fr
3 **Microsoft Research - INRIA Joint Centre, Palaiseau, France**
laurent.massoulie@inria.fr

Abstract

Motivated by community detection, we characterise the spectrum of the non-backtracking matrix B in the Degree-Corrected Stochastic Block Model.

Specifically, we consider a random graph on n vertices partitioned into two asymptotically equal-sized clusters. The vertices have i.i.d. weights $\{\phi_u\}_{u=1}^n$ with second moment $\Phi^{(2)}$. The intra-cluster connection probability for vertices u and v is $\frac{\phi_u\phi_v}{n}a$ and the inter-cluster connection probability is $\frac{\phi_u\phi_v}{n}b$.

We show that with high probability, the following holds: The leading eigenvalue of the non-backtracking matrix B is asymptotic to $\rho = \frac{a+b}{2}\Phi^{(2)}$. The second eigenvalue is asymptotic to $\mu_2 = \frac{a-b}{2}\Phi^{(2)}$ when $\mu_2^2 > \rho$, but asymptotically bounded by $\sqrt{\rho}$ when $\mu_2^2 \leq \rho$. All the remaining eigenvalues are asymptotically bounded by $\sqrt{\rho}$. As a result, a clustering positively-correlated with the true communities can be obtained based on the second eigenvector of B in the regime where $\mu_2^2 > \rho$.

In a previous work we obtained that detection is impossible when $\mu_2^2 < \rho$, meaning that there occurs a phase-transition in the sparse regime of the Degree-Corrected Stochastic Block Model.

As a corollary, we obtain that Degree-Corrected Erdős-Rényi graphs asymptotically satisfy the graph Riemann hypothesis, a quasi-Ramanujan property.

A by-product of our proof is a weak law of large numbers for local-functionals on Degree-Corrected Stochastic Block Models, which could be of independent interest.

1998 ACM Subject Classification E.1 Graphs and Networks

Keywords and phrases Degree-Corrected Stochastic Block Model, Non-backtracking Matrix, Ramanujan Graphs

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.44

1 Introduction

The non-backtracking matrix B of a graph $G = (V, E)$ is indexed by the set of its oriented edges $\vec{E} = \{(u, v) : \{u, v\} \in E\}$. For $e = (e_1, e_2), f = (f_1, f_2) \in \vec{E}$, B is defined as

$$B_{ef} = 1_{e_2=f_1} 1_{e_1 \neq f_2}.$$

This matrix was introduced by Hashimoto [10] in 1989.

We study the spectrum of B when G is a random graph generated according to the Degree-Corrected Stochastic Block Model (DC-SBM) [11]. We characterise its leading eigenvalues

© Lennart Gulikers, Marc Lelarge, and Laurent Massoulié;
licensed under Creative Commons License CC-BY

and corresponding eigenvectors when the number of vertices in G tends to infinity. Our motivation stems from community detection problems: experiments in [14] show that the spectral method based on the non-backtracking matrix seems to work well on real datasets. We test the robustness of this method and show in particular that, above a certain threshold, the second eigenvector of B is correlated with the underlying communities.

The DC-SBM [11] is an extension of the *ordinary* Stochastic Block Model (SBM) [8]. The latter model has as a drawback that vertices in the same community are stochastically indistinguishable and it therefore fails to accurately describe networks with high heterogeneity. Compare this to fitting a straight line on intrinsically curved data, which is doomed to miss important information. The DC-SBM is a more realistic model: it allows for very general degree-sequences.

The special case of the DC-SBM under consideration here is defined as follows: It is a random graph on n vertices partitioned into two asymptotically equal-sized clusters. The vertices have bounded i.i.d. weights $\{\phi_u\}_{u=1}^n$ with second moment $\Phi^{(2)}$. The intra-cluster connection probability for vertices u and v is $\frac{\phi_u\phi_v}{n}a$ and the inter-cluster connection probability is $\frac{\phi_u\phi_v}{n}b$, for two constants $a, b > 0$.

Note that those graphs are thus sparse, which is a challenging regime for community detection. Indeed, in the *ordinary* SBM (obtained by putting $\phi_1 = \ldots = \phi_n = 1$), an instance of the graph might not contain enough information to distinguish between the two clusters if the difference between a and b is small. More precisely, reconstruction is impossible when $(a-b)^2 \leq 2(a+b)$ [18]. Interestingly, positively-correlated reconstruction can be obtained by thresholding the second-eigenvector of B [2] immediately above the threshold (i.e., $(a-b)^2 > 2(a+b)$). The SBM thus has a phase-transition in its sparse regime.

Does the DC-SBM exhibit a similar behaviour? We showed in an earlier work [6] that detection is impossible when $(a-b)^2\Phi^{(2)} \leq 2(a+b)$. In our current work we analyse the regime where $(a-b)^2\Phi^{(2)} > 2(a+b)$. We answer the following questions: is detection possible in this regime and if so, can we use again the non-backtracking matrix or do we need to modify it? *A priori this is unclear, because an algorithm solely based on B cannot use any information on the weights as input.* Our main result shows that the spectral method based on the non-backtracking matrix (thus the same method as in [2]) successfully detects communities in the regime $(a-b)^2\Phi^{(2)} > 2(a+b)$. Surprisingly, no modification of the matrix, nor information about the weights is needed (compare this to the adjacency matrix, which needs to be adapted to the degree-corrected setting [7]), which shows the robustness of the method. Moreover as in the standard SBM, the algorithm is optimal in the sense that it works all the way down to the detectability-threshold.

Informally, we have the following results: With high probability, the leading eigenvalue of the non-backtracking matrix B is asymptotic to $\rho = \frac{a+b}{2}\Phi^{(2)}$. The second eigenvalue is asymptotic to $\mu_2 = \frac{a-b}{2}\Phi^{(2)}$ when $\mu_2^2 > \rho$, but asymptotically bounded by $\sqrt{\rho}$ when $\mu_2^2 \leq \rho$. All the remaining eigenvalues are asymptotically bounded by $\sqrt{\rho}$. Further, a clustering positively-correlated with the true communities can be obtained based on the second eigenvector of B in the regime where $\mu_2^2 > \rho$ (i.e., precisely when $(a-b)^2\Phi^{(2)} > 2(a+b)$).

A side-result is that Degree-Corrected Erdős-Rényi graphs asymptotically satisfy the graph Riemann hypothesis, a quasi-Ramanujan property.

In our proof we derive and use a weak law of large numbers for local-functionals on Degree-Corrected Stochastic Block Models, which could be of independent interest.

1.1 Community detection background

In this paper we are interested in community detection: The problem of clustering vertices in a graph into groups of "similar" nodes. In particular, the graphs here are generated according to the DC-SBM and the goal is to retrieve the spin (or group-membership) of the nodes based on a single observation of the DC-SBM.

When the average degree of a vertex grows sufficiently fast with the size of the network (i.e., the average degree is $\Omega(\log(n))$), we speak about dense networks. Community-detection is then well understood and we consider instead sparse graphs where the average degree is bounded by a constant. This setting is more realistic as most real networks are sparse, but is at the same time more challenging. Indeed, traditional methods based on the Adjacency or Laplacian matrix working well in the dense case break down when employed in the sparse case.

In the sparse regime, with high probability, at least a positive fraction of the nodes is isolated. Consequently, one cannot hope to find the community-membership of *all* vertices. We therefore address here the problem of finding a clustering that is positively correlated with the true community-structure.

In [3] it was first conjectured that a detectability phase transition exists in the *ordinary* SBM: When $(a-b)^2 > 2(a+b)$, the belief propagation algorithm would succeed in finding such a positively correlated clustering. Conversely, due to a lack of information, detection would be impossible when $(a-b)^2 \leq 2(a+b)$.

In [18], impossibility of reconstruction when $(a-b)^2 \leq 2(a+b)$ is shown for the SBM. This paper builds further on a tree-reconstruction problem in [4].

The authors of [14] conjectured that detection using the second eigenvector of B would succeed all the way down to the conjectured detectability threshold. Two variants of this so-called spectral redemption conjecture were proven before the work in [2] appeared:

In [16] it is shown that detection based on the second eigenvector of a matrix counting self-avoiding paths in the graph leads to consistent recovery when $\left(\frac{a-b}{2}\right)^2 > \frac{a+b}{2}$.

Independently, in [17], the authors prove the positive side of the conjecture by using a constructing based on counting non-backtracking paths in graphs generated according to the SBM.

More recently, in [2] the spectral redemption conjecture is proved. This work moreover determines the limits of community detection based on the non-backtracking spectrum in the presence of an arbitrary number of communities.

Here we extend the work in [2] to the more general setting of the DC-SBM.

1.2 Quasi Ramanujan property

Following the definition introduced in [15], a k-regular graph is Ramanujan if its second largest absolute eigenvalue is no larger than $2\sqrt{k-1}$. In [9], a graph is said to satisfy the graph Riemann hypothesis if B has no eigenvalues λ such that $|\lambda| \in (\sqrt{\rho_B}, \rho_B)$, where ρ_B is the Perron-Frobenius eigenvalue of B. The graph Riemann hypothesis can be seen as a generalization of the Ramanujan property, because a regular graph satisfies the graph Riemann hypothesis if and only if it has the Ramanujan property [9, 19].

Now, put $a = b = 1$ to obtain a Degree-Corrected Erdős-Rényi graph where vertices u and v are connected by an edge with probability $\frac{\phi_u \phi_v}{n}$. Our results imply that, with high probability, $\rho_B = \Phi^{(2)} + o(1)$, while all other eigenvalues are in absolute value smaller than $\sqrt{\Phi^{(2)}} + o(1)$. Consequently, these Degree-Corrected Erdős-Rényi graphs asymptotically satisfy the graph Riemann hypothesis.

1.3 Outline and main differences with ordinary SBM

We follow the same general approach as in [2]. We focus primarily on the differences and complications here: we often omit or shorten the proof of a statement if it may be proven in a very similar way.

In Section 2 we define the DC-SBM and state the assumptions we make. This is then followed by Theorem 1 on the spectrum of B and its consequences for community detection, Theorem 2.

In Section 3, we give the necessary background on non-backtracking matrices. Further, we give an extension of the Bauer-Fike Theorem, that first appeared in [2].

In Section 4 we give the proof of Theorem 1. It builds on Propositions 4 and 5. Their proofs are deferred to later sections.

In Section 5 we consider two-type branching process where the offspring distribution is governed by a Poisson mixture to capture the weights of the vertices. We associate two martingales to this process and extend limiting results by Kesten and Stigum [12, 13]. Hoeffding's inequality plays an important role here to prove concentrations results for the weights. Further, we define a cross-generational functional on these branching processes that is correlated with the spin of the root.

In Section 6 we state a coupling between local neighbourhoods and the branching process with weights in Section 5. We established this coupling in an earlier work [6], it is technically more involved than the ordinary coupling on graphs with unit weight. It is crucial that the weights in the graph and the branching process are perfectly coupled. We further establish a growth condition on the local neighbourhoods, using a stochastic domination argument that is more involved than its analogue in unweighed graphs.

In Section 7 we define local functionals that map graphs, together with their spins *and* weights to the real numbers. We establish, using Efron-Stein's inequality, a weak law of large numbers for those functionals, which could be of independent interest. Part of the work here is again hidden in the coupling from [6].

In Section 8 we apply those local functionals to establish Proposition 4.

In Section 9 we decompose powers of the matrix B as a sum of products. This technique appeared first in [16] for matrices counting self-avoiding paths and was elaborated in [2]. To bound the norm of the individual matrices occurring in the decomposition, we use the trace method initiated in [5]. In doing so, we need to bound the expectation of products of higher moments of the weights over certain paths. This is a significant complication with respect to the ordinary SBM, see Section 9.2 for a comparison.

In Section 10 we prove that positively correlated clustering is possible based on the second eigenvector of B, i.e., Theorem 2. We use the symmetry present in the two-communities setting here, which gets in general broken in models with more than two communities.

Detailed proofs of the statements in Sections 5, 6, 7, 9 and 10 can be found in Appendices A–E in the detailed version of the underlying article: arXiv:1609.02487.

In each section we give a detailed comparison with the ordinary SBM.

2 Main Results

We define our model more precisely and state the two main theorems.

We consider random graphs on n nodes $V = \{1, \dots, n\}$ drawn according to the Degree-Corrected Stochastic Block Model [11]. The vertices are partitioned into two clusters of sizes n_+ and n_- by giving each vertex v a spin $\sigma(v)$ from $\{+, -\}$. The vertices have i.i.d. weights

$\{\phi_u\}_{u=1}^n$ governed by some law ν with support in $[\phi_{\min}, \phi_{\max}]$, where $0 < \phi_{\min} \leq \phi_{\max} < \infty$ are constants. We denote the second moment of the weights by $\Phi^{(2)}$. An edge is drawn between nodes u and v with probability $\frac{\phi_u\phi_v}{n}a$ when u and v have the same spin and with probability $\frac{\phi_u\phi_v}{n}b$ otherwise. The model parameters a and b are constant. We assume that for some constant $\gamma \in (0, 1]$,

$$n_\pm = \frac{n}{2} + \mathcal{O}(n^{1-\gamma}), \tag{1}$$

i.e., the communities have nearly equal size.

The ordinary SBM on two or more communities was first introduced in [8], which is a generalization of Erdős-Rényi graphs. The Degree-Corrected SBM appeared first in [11]. General inhomogeneous random graphs are considered in [1].

Note that we retrieve the two-communities ordinary SBM by giving all nodes *unit* weight.

Local neighbourhoods in the sparse graphs under consideration are tree-like with high probability. In [6] we showed that these trees are distributed according to a Poisson-mixture two-type branching process, detailed in Section 5 below. We denote the mean progeny matrix of the branching process by

$$M = \frac{\Phi^{(2)}}{2}\begin{pmatrix} a & b \\ b & a \end{pmatrix}. \tag{2}$$

We introduce the orthonormal vectors

$$g_1 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \text{ and } g_2 = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}, \tag{3}$$

together with the scalars

$$\rho = \mu_1 = \frac{a+b}{2}\Phi^{(2)}, \text{ and } \mu_2 = \frac{a-b}{2}\Phi^{(2)}. \tag{4}$$

Then, g_k $(k = 1, 2)$ are the left-eigenvectors of M associated to eigenvalues μ_k:

$$g_k^* M = \mu_k g_k^*, \quad k = 1, 2. \tag{5}$$

Note that ρ and μ_2 are also asymptotically eigenvalues of the expected adjacency matrix conditioned on the weights.

Indeed, if A denotes the adjacency matrix, and if ψ_1 and ψ_2 are the vectors defined for $u \in V$ by $\psi_1(u) = \frac{1}{\sqrt{2}}\phi_u$ and $\psi_2(u) = \frac{1}{\sqrt{2}}\sigma_u\phi_u$, then

$$\mathbb{E}\left[A|\phi_1, \ldots, \phi_n\right] = \frac{a+b}{n}\psi_1\psi_1^* + \frac{a-b}{n}\psi_2\psi_2^* - a\frac{1}{n}\mathrm{diag}\{\phi_u^2\}.$$

Put $\widehat{\psi}_i = \frac{\psi_i}{\|\psi_i\|_2}$. Then, by the law of large numbers, for $i = 1, 2$,

$$\left\|\mathbb{E}\left[A|\phi_i, \ldots, \phi_n\right]\widehat{\psi}_i - \mu_i\widehat{\psi}_i\right\|_2 \to 0,$$

in probability, as n tends to ∞.

Finally, we define for $k \in \{1, 2\}$,

$$\chi_k(e) = g_k(\sigma(e_2))\phi_{e_2}, \quad \text{for } e \in \vec{E}. \tag{6}$$

We show that the *candidate eigenvectors*

$$\zeta_k = \frac{B^\ell B^{*\ell}\check{\chi}_k}{\|B^\ell B^{*\ell}\check{\chi}_k\|} \tag{7}$$

are then, for $\ell \sim \log(n)$, asymptotically aligned with the first two eigenvectors of B. Note the weight in (6), which is *not* present in the ordinary SBM.

▶ **Theorem 1** (Degree-Corrected Extension of Theorem 4 in [2]). *Let G be drawn according to the DC-SBM such that assumption (1) holds. Assume that $\ell = C_{min}\log(n)$, with $C_{min} > 0$ a small constant defined in (9).*

If $\mu_2^2 > \rho$, then, with high probability, the eigenvalues λ_i of B satisfy

$$|\lambda_1 - \rho| = o(1), |\lambda_2 - \mu_2| = o(1), \quad \text{and, for } i \geq 3, \quad |\lambda_i| \leq \sqrt{\rho} + o(1).$$

Further, if, for $k \in \{1,2\}$, ξ_k is a normalized eigenvector associated to λ_k, then ξ_k is asymptotically aligned with ζ_k. The vectors ξ_1 and ξ_2 are asymptotically orthogonal.

If $\rho > 1$, and $\mu_2^2 \leq \rho$, then, with high probability, the eigenvalues λ_i of B satisfy

$$|\lambda_1 - \rho| = o(1), \quad \text{and, for } i \geq 2, \quad |\lambda_i| \leq \sqrt{\rho} + o(1).$$

Further, ξ_1 is asymptotically aligned with ζ_1.

Note that $\mu_2^2 > \rho$ implies $\rho > 1$, so that we consider the DC-SBM precisely in the regime where a giant component emerges, see [1].

In Theorem 2 we show that positively correlated clustering is possible based on the second eigenvector of B when above the feasibility threshold. More precisely, let $\widehat{\sigma} = \{\widehat{\sigma}(v)\}_{v\in V}$ be estimators for the spins of the vertices. Following [3], we say that $\widehat{\sigma}$ has positive overlap with the true spin configuration $\sigma = \{\sigma(v)\}_{v\in V}$ if for some $\delta > 0$, with high probability,

$$\min_p \frac{1}{n}\sum_{v=1}^{n} 1_{\widehat{\sigma}(v)=p\circ\sigma(v)} > \frac{1}{2} + \delta,$$

where p runs over the identity mapping on $\{+,-\}$ and the permutation that swaps $+$ and $-$.

▶ **Theorem 2** (Degree-Corrected Extension of Theorem 5 in [2]). *Let G be drawn according to the DC-SBM such that assumption (1) holds and such that $\mu_2^2 > \rho$. Let ξ_2 be the second normalized eigenvector of B.*

Then, there exists a deterministic threshold $\tau \in \mathbb{R}$, such that the following procedure yields asymptotically positive overlap: Put for vertex $v \in V$ its estimator $\widehat{\sigma}(v) = +$ if $\sum_{e:e_2=v} \xi_2(e) > \frac{\tau}{\sqrt{n}}$ and put $\widehat{\sigma}(v) = -$ otherwise.

2.1 Notation

We say that a sequence $(E_n)_n$ of events happens with high probability (w.h.p.) if $\lim_{n\to\infty} \mathbb{P}(E_n) = 1$.

We denote by $\|\cdot\|$ both the euclidean norm for vectors and the operator norm of matrices. I.e., for vectors $x = (x_1, \ldots, x_m)$, and a matrix A, $\|x\| = \sqrt{\sum_{u=1}^m x_u^2}$, and $\|A\| = \sup_{x,\|x\|=1} \|Ax\|$.

Below we use that the neighbourhoods with a radius no larger than $C_{\text{coupling}}\log_\rho(n)$ can be coupled w.h.p. to certain branching processes, where

$$C_{\text{coupling}} := \frac{\left(\frac{1}{3} - \frac{1}{9}\log(4/e)\right) \wedge \left(\frac{1}{80} \wedge \frac{\gamma}{4}\right)}{\log_\rho(2(a+b)\phi_{\max}^2)}. \tag{8}$$

We put,

$$C_{\min} = \frac{1}{10} C_{\text{coupling}} \tag{9}$$

and consider often neighbourhoods of radius $C_{\min}\log_\rho(n)$.

We denote the k-th moment of the weight distribution ν by $\Phi^{(k)}$. I.e., $\mathbb{E}\left[\phi_1^k\right] = \Phi^{(k)}$.

The non-backtracking property for oriented edges $e, f \in \vec{E}$ is denoted by $e \to f$, i.e., $e_2 = f_1$ and $f_2 \neq e_1$.

In proofs, we often use the symbols $c_1, c_2, \ldots$ for suitably chosen constants.

3 Preliminaries

3.1 Background on non-backtracking matrix

We repeat here the most important observations made in [2].

Firstly, for any $k \geq 1$, B^k_{ef} counts the number of non-backtracking paths between oriented edges e and f. A non-backtracking path is defined as an oriented path between two oriented edges such that no edge is the inverse of its preceding edge, i.e., the path makes no backtrack.

Another import observation is that $(B^*)_{ef} = B_{fe} = B_{e^{-1}f^{-1}}$, where for oriented edge $e = (e_1, e_2)$, we set $e^{-1} = (e_2, e_1)$. If we introduce the swap notation, for $x \in \mathbb{R}^{\vec{E}}$,

$$\check{x}_e = x_{e^{-1}}, \quad e \in \vec{E},$$

then for any $x, y \in \mathbb{R}^{\vec{E}}$, and integer $k \geq 0$,

$$\langle y, B^k x \rangle = \langle B^k \check{y}, \check{x} \rangle.$$

Denote by P the matrix on $\mathbb{R}^{\vec{E} \times \vec{E}}$, defined on oriented edges e, f as

$$P_{ef} = 1_{f=e^{-1}}.$$

Then, $Px = \check{x}$, $P^* = P$ and $P^{-1} = P$. Further,

$$(B^k P)^* = P(B^*)^k = B^k P,$$

so that we can write the symmetric matrix $B^k P$ in diagonal form: Let $(\sigma_{k,j})_j$ be eigenvalues of $B^k P$ ordered in decreasing order of absolute value, and let $(x_{k,j})_j$ be the corresponding orthonormal eigenvectors. Then,

$$B^k = (B^k P)P = \sum_j \sigma_{k,j} x_{k,j} x^*_{k,j} P = \sum_j \sigma_{k,j} x_{k,j} \check{x}^*_{k,j} = \sum_j s_{k,j} x_{k,j} y^*_{k,j}, \tag{10}$$

where $s_{k,j} = |\sigma_{k,j}|$ and $y_{k,j} = \mathrm{sign}(\sigma_{k,j}) \check{x}_{k,j}$. Since P is an orthogonal matrix, $(\check{x}_{k,j})_j$ form an orthonormal base for $\mathbb{R}^{\vec{E}}$ and the term furthest on the right of (10) is thus the spectral value decomposition of B^k. Now, if B is irreducible and if ξ denotes the normalized Perron eigenvector of B with eigenvalue $\lambda_1(B) > 0$, we have $\lambda_1(B) = \lim_{k\to\infty} (\sigma_{k,1})^{1/k}$, and $\lim_{k\to\infty} \|x_{k,1} - \xi\| = 0$.

In [2], the Bauer-Fike Theorem is extended to prove the spectral claims we make here.

3.2 Extension of Bauer-Fike Theorem

Tailored to our needs, we use the following proposition from [2]:

▶ **Proposition 3** (Special case of Proposition 8 in [2]). *Let $\ell = C \log_\rho n$, with $C > 0$. Let $A \in M_n(\mathbb{R})$, such that for some vectors $x_1 = x_{\ell,1}, y_1 = y_{\ell,1}, x_2 = x_{\ell,2}, y_2 = y_{\ell,2} \in \mathbb{R}$, some matrix $R_\ell \in M_n(\mathbb{R})$, and some non-zero constants $\rho > \mu_2$ with $\mu_2^2 > \rho$,*

$$A^\ell = \rho^\ell x_1 y_1^* + \mu_2^\ell x_2 y_2^* + R_\ell. \tag{11}$$

Assume there exist $c_0, c_1 > 0$ such that for all $i \in \{1, 2\}$, $\langle y_i, x_i \rangle \geq c_0$, $\|x_i\| \|y_i\| \leq c_1$. Assume further that $\langle x_1, y_2 \rangle = \langle x_2, y_1 \rangle = \langle x_1, x_2 \rangle = \langle y_1, y_2 \rangle = 0$ and for some $c > 0$

$$\|R_\ell\| < \rho^{\ell/2} \log^c(n).$$

Let $(\lambda_i)_{1\le i\le n}$, be the eigenvalues of A with $|\lambda_n|\le\ldots\le|\lambda_1|$. Then,

$$|\lambda_1-\rho|=o(1),|\lambda_2-\mu_2|=o(1),\quad \text{and, for } i\ge 3,\quad |\lambda_i|\le\sqrt{\rho}+o(1).$$

Further, there exist unit eigenvectors ψ_1,ψ_2 of A with eigenvalues λ_1, respectively λ_2 such that

$$||\psi_i-\frac{x_i}{\|x_i\|}||=o(1).$$

Proof. This is a special case of Proposition 8 in [2]. In the notation of the latter, we have $\ell'=\ell-2$, $\theta_1=\rho$, $\theta_2=\mu_2$, $\theta=\mu_2$, $\gamma\ge\frac{a+b}{|a-b|}>1$. Further $\frac{c_0(c_0\gamma^k-c_1)_+}{4c_1}\wedge\frac{c_0^2}{2(\ell\vee\ell')c_1}\sim\frac{1}{\log_\rho n}$, and thus

$$\|R_\ell\|\le\log^c(n)\left(\frac{\sqrt{\rho}}{|\mu_2|}\right)^\ell|\mu_2|^\ell=o(1)\frac{1}{\log_\rho n}|\theta|^\ell.$$ ◀

To prove the case $\mu_2^2>\rho$ of Theorem 1, we thus need to find candidate vectors x_1,x_2,y_1 and y_2 that meet the conditions in Proposition 3 and further verify that the remainder R_ℓ has small norm. Note that the last condition is true whenever $\|B^\ell x\|\le\rho^{\ell/2}\log^c(n)$ for all normalized x in $\mathrm{span}\{y_1,y_2\}^\perp$.

To address the case $\mu_2^2\le\rho$ of Theorem 1, we appeal to Proposition 7 in [2], which is very similar in spirit to Proposition 3.

4 Proof of Theorem 1

4.1 The case $\mu_2^2>\rho$

We start with the case $\mu_2^2>\rho$. We decompose, for some vectors x_1,y_1,x_2 and y_2 and matrix R_ℓ,

$$B^\ell=\rho^\ell x_1y_1^*+\mu_2^\ell x_2y_2^*+R_\ell,$$

and we show that the assumptions of Proposition 3 are met.

Let ℓ be as in Theorem 1 and recall χ_k and ζ_k from (6) and (7). For ease of notation, we introduce for $k\in\{1,2\}$,

$$\varphi_k=\frac{B^\ell\chi_k}{\|B^\ell\chi_k\|},\quad \text{and } \theta_k=\|B^\ell\check{\varphi}_k\|. \tag{12}$$

Then, $\zeta_k=\frac{B^\ell\check{\varphi}_k}{\theta_k}$.

To prove the main theorem, we need the following two propositions. The proofs are deferred to Section 8 and 9.1. The material in Section 8 builds on ingredients from Sections 6 - 7, where we assume that $\mu_2^2>\rho$, unless stated otherwise.

▶ **Proposition 4** (Degree-Corrected Extension of Proposition 19 in [2]). *Assume that $\mu_2^2>\rho$. Let $\ell=C\log_\rho n$ with $0<C<C_{min}$. For some $b,c>0$, with high probability,*

(i) $b|\mu_k^\ell|\le\theta_k\le c|\mu_k^\ell|$ *if* $k\in\{1,2\}$,
(ii) $sign(\mu_k^\ell)\langle\zeta_k,\check{\varphi}_k\rangle\ge b$ *if* $k\in\{1,2\}$,
(iii) $|\langle\varphi_1,\varphi_2\rangle|\le(\log n)^3n^{C-\left(\frac{\gamma}{2}\wedge\frac{1}{40}\right)}$,
(iv) $|\langle\zeta_j,\check{\varphi}_k\rangle|\le(\log n)^3n^{\frac{3}{2}C-\left(\frac{\gamma}{2}\wedge\frac{1}{40}\right)}$ *if* $k\ne j\in\{1,2\}$.
(v) $|\langle\zeta_1,\zeta_2\rangle|\le(\log n)^8n^{2C-\left(\frac{\gamma}{2}\wedge\frac{1}{40}\right)}$.

Put $H = \text{span}\{\check{\varphi}_1, \check{\varphi}_2\}$, then

▶ **Proposition 5** (Degree-Corrected Extension of Proposition 20 in [2]). *Let $\ell = C \log_\rho n$ with $0 < C < C_{min}$. For some $c > 0$, with high probability,*

$$\sup_{x \in H^\perp, \|x\|=1} \|B^\ell x\| \leq (\log n)^c \rho^{\ell/2}. \tag{13}$$

Put $\bar{\varphi}_1 = \check{\varphi}_1$, and $\bar{\varphi}_2 = \frac{\check{\varphi}_2 - \langle \check{\varphi}_1, \check{\varphi}_2 \rangle \check{\varphi}_1}{||\check{\varphi}_2 - \langle \check{\varphi}_1, \check{\varphi}_2 \rangle \check{\varphi}_1||}$, then $\bar{\varphi}_1$ and $\bar{\varphi}_2$ are orthonormal and $||\bar{\varphi}_2 - \check{\varphi}_2|| = o(\rho^{-\ell/2})$, due to Proposition 4 (iii).

Let $\bar{\zeta}_1$ be the normalized orthogonal projection of ζ_1 on $\text{span}\{\bar{\varphi}_2\}^\perp$. Similarly, let $\bar{\zeta}_2$ be the normalized orthogonal projection of ζ_2 on $\text{span}\{\bar{\zeta}_1, \bar{\varphi}_1\}^\perp$.

Then $\langle \bar{\zeta}_1, \bar{\zeta}_2 \rangle = 0$ and for $i = 1, 2$, $||\bar{\zeta}_i - \zeta_i|| = o(\rho^{-\ell/2})$, as follows from Proposition 4 (iv) and (v).

We set

$$D = \theta_1 \bar{\zeta}_1 \bar{\varphi}_1^* + \theta_2 \bar{\zeta}_2 \bar{\varphi}_2^* = \rho^\ell \left(\frac{\theta_1}{\rho^\ell} \bar{\zeta}_1 \right) \bar{\varphi}_1^* + \mu_2^\ell \left(\frac{\theta_2}{\mu_2^\ell} \bar{\zeta}_2 \right) \bar{\varphi}_2^*.$$

Note that,

$$\|B^\ell \bar{\varphi}_1\| = \theta_1 = O(\rho^\ell),$$

and

$$\|B^\ell \bar{\varphi}_2\| = \|B^\ell \left((1 + o(1)) \check{\varphi}_2 + o(1) \bar{\varphi}_1 \right) \| = O(\rho^\ell).$$

As a consequence, from Proposition 5,

$$\|B^\ell\| = O(\rho^\ell).$$

Since $D\bar{\varphi}_i = B^\ell \check{\varphi}_i + \theta_i (\bar{\zeta}_i - \zeta_i)$,

$$\|B^\ell \bar{\varphi}_i - D\bar{\varphi}_i\| \leq \|B^\ell\| \|\bar{\varphi}_i - \check{\varphi}_i\| + \theta_i \|\bar{\zeta}_i - \zeta_i\| = \mathcal{O}\left(\rho^{\ell/2}\right).$$

Let P be the orthogonal projection on $H = \text{span}\{\bar{\varphi}_1, \bar{\varphi}_2\} = \text{span}\{\check{\varphi}_1, \check{\varphi}_2\}$, then $\|B^\ell P - D\| = \mathcal{O}\left(\rho^{\ell/2}\right)$.

Put $R_\ell = B^\ell - D$. Write for $y \in \mathbb{R}^{\vec{E}}$ with unit norm, $y = h + h^\perp$, with $h \in H$ and $h^\perp \in H^\perp$, then

$$\begin{aligned} \|R_\ell y\| &= \|B^\ell h^\perp + (B^\ell - D)h\| \\ &\leq \sup_{x \in H^\perp, \|x\|=1} \|B^\ell x\| + \|B^\ell P - D\| \\ &= \mathcal{O}\left(\log^c(n) \rho^{\ell/2}\right), \end{aligned} \tag{14}$$

as follows from Proposition 5.

We finish by applying Proposition 3 with $x_1 = \frac{\theta_1}{\rho^\ell} \bar{\zeta}_1, y_1 = \bar{\varphi}_1, x_2 = \frac{\theta_2}{\mu_2^\ell} \bar{\zeta}_2$, and, $y_2 = \bar{\varphi}_2$.

4.2 The case $\mu_2^2 \leq \rho$

In case $\mu_2^2 \leq \rho$, Proposition 4 (i) and (ii) continue to hold for $k = 1$. Further, Proposition 4 (iii) as well as Proposition 5 continue to hold. We need however the following bound for $k = 2$:

▶ **Proposition 6.** *Assume that* $\mu_2^2 \le \rho$. *Let* $\ell = C \log_\rho n$ *with* $0 < C < C_{min}$. *For some* $c > 0$, *with high probability,*

$$\theta_2 \le (\log n)^c \rho^{\ell/2}.$$

Using this proposition and $||\bar{\varphi}_2 - \check{\varphi}_2|| = o(\rho^{-\ell/2})$, we get

$$\|B^\ell \bar{\varphi}_2\| \le (\log n)^{c+1} \rho^{\ell/2}.$$

It remains to apply Proposition 7 from [2].

5 Poisson-mixture two-type branching processes

The proofs of the statements in this section can be found in Appendix A in the detailed version of the underlying article (Arxiv:1609.02487).

5.1 A theorem of Kesten and Stigum

We consider the following branching process starting with a single particle, the root o, having spin $\sigma_o \in \{+,-\}$ and weight $\phi_o \in [\phi_{\min}, \phi_{\max}]$ (which we often take random). The root is replaced in generation 1 by Poi $\left(\frac{a}{2}\Phi^{(1)}\phi_o\right)$ particles of spin σ_o and Poi $\left(\frac{b}{2}\Phi^{(1)}\phi_o\right)$ particles of spin $-\sigma_o$. Further, the weights of those particles are i.i.d. distributed following law ν^*, the size-biased version of ν, defined for $x \in [\phi_{\min}, \phi_{\max}]$ by

$$\nu^*([0,x]) = \frac{1}{\Phi^{(1)}} \int_{\phi_{\min}}^{x} y \mathrm{d}\nu(y). \tag{15}$$

For generation $t \ge 1$, a particle with spin σ and weight ϕ^* is replaced in the next generation by Poi $\left(\frac{a}{2}\Phi^{(1)}\phi^*\right)$ particles with the same spin and Poi $\left(\frac{b}{2}\Phi^{(1)}\phi^*\right)$ particles of the opposite sign. Again, the weights of the particles in generation $t+1$ follow in an i.i.d. fashion the law ν^*. The offspring-size of an individual is thus a **Poisson-mixture**.

We use the notation $Z_t = \begin{pmatrix} Z_t(+) \\ Z_t(-) \end{pmatrix}$ for the population at generation $t \ge 1$, where $Z_t(\pm)$ is the number of type $\pm$ particles in generation t. We let $(\mathcal{F}_t)_{t\ge1}$ denote the natural filtration associated to $(Z_t)_{t\ge1}$.

We associate two matrices to the branching process, namely M defined in (2), and, for a root with weight ϕ_o,

$$M_{\phi_o} = \frac{\Phi^{(1)}\phi_o}{\Phi^{(2)}} M. \tag{16}$$

Then, M is the *transition* matrix for generations $t \ge 1$ and later:

$$\mathbb{E}\left[Z_{t+1} | Z_t\right] = M Z_t, \quad \text{for all } t \ge 1, \tag{17}$$

and M_{ϕ_o} describes the transition from the root to the first generation:

$$\mathbb{E}\left[Z_1 | Z_0, \phi_o\right] = M_{\phi_o} Z_0, \tag{18}$$

where, by assumption $Z_0 = \begin{pmatrix} 1_{\sigma_o=+} \\ 1_{\sigma_o=-} \end{pmatrix}$. Note that the difference between the root and later generations stems from the fact that the root's weight is deterministic in the conditional expectation, whereas the weight of a particle in any later generation has expectation $\frac{\Phi^{(2)}}{\Phi^{(1)}}$.

Recall from (5) that g_k $(k = 1, 2)$ are the left-eigenvectors of M associated to eigenvalues μ_k:

$$g_k^* M = \mu_k g_k^*, \quad k = 1, 2. \tag{19}$$

Note that M_{ϕ_o} has the same left-eigenvectors as M, while the corresponding eigenvalues are given by

$$\mu_{k,\phi_o} = \frac{\Phi^{(1)}\phi_o}{\Phi^{(2)}}\mu_k, \quad k = 1, 2. \tag{20}$$

Theorem 7 shows that a Kesten-Stigum theorem applies to the "classical" branching process obtained after restricting the above process to generations 1 and later. Corollary 8, then, joins this classical branching process to the transition from the root to generation 1.

We further consider the vector $\Psi_t = (\Psi_t(+), \Psi_t(-))$, containing sums of the weights,

$$\Psi_t(\pm) = \sum_{u \in Y_t} 1_{\sigma_u = \pm}\phi_u, \tag{21}$$

where Y_t is the set of particles at distance t from the root, and where ϕ_u and σ_u denote the weight respectively spin of a particle u. Note that $\Psi_t = Z_t$ in case of unit weights.

The martingale Theorem 9 is not present in [2]. We need it to bound the variance of the cross-generational functional defined in Section 5.3.

▶ **Theorem 7** (Degree-Corrected Extension of Theorem 21 in [2]). *Assume that $\mu_2^2 > \rho$. Put $\mathcal{F}_t = \{Z_s\}_{s \le t}$. For any $k = 1, 2$,*

$$\left(X_k(t) := \frac{\langle g_k, Z_t \rangle}{\mu_k^{t-1}} - \langle g_k, Z_1 \rangle \right)_{t \ge 1},$$

is an $\mathcal{F}_t$-martingale converging a.s. and in L^2 such that for some $C > 0$ and all $t \ge 1$, $\mathbb{E}[X_k(t)] = 0$ and $\mathbb{E}\left[X_k^2(t)|Z_1\right] \le C\|Z_1\|_1$.

▶ **Corollary 8.** *Assume that $\mu_2^2 > \rho$. For $k = 1, 2$, with the weight $\phi_o = \psi_o$ of the root fixed, the sequence of random variables $(Y_{k,\psi_o}(t))_{t \ge 1} = \left(\frac{\langle g_k, Z_t \rangle}{\mu_k^{t-1}\mu_{k,\psi_o}}\right)_{t \ge 1}$ converges* almost surely *and in L^2 to a random variable $Y_{k,\psi_o}(\infty)$ with $\mathbb{E}[Y_{k,\psi_o}(\infty)|\sigma_o] = g_k(\sigma_o)$. Further, the L^2-convergence takes place uniformly over all ψ_o.*

▶ **Theorem 9.** *Assume that $\mu_2^2 > \rho$. Put $\mathcal{G}_t = \{\Psi_s\}_{s \le t}$. For any $k = 1, 2$,*

$$\left(X_k(t) := \frac{\langle g_k, \Psi_t \rangle}{\mu_k^{t-1}} - \langle g_k, \Psi_1 \rangle \right)_{t \ge 1},$$

is an $\mathcal{G}_t$-martingale converging a.s. and in L^2 such that for some $C > 0$ and all $t \ge 1$, $\mathbb{E}[X_k(t)] = 0$ and $\mathbb{E}\left[X_k^2(t)|Z_1\right] \le C\|Z_1\|_1$.

5.2 Quantitative version of the Kesten-Stigum theorem

We now quantify the growth of the population size. The latter is defined as

$$S_t = \|Z_t\|_1, \quad t \ge 0,$$

i.e., the number of individuals in generation $t \ge 0$. Given S_t, for $t \ge 1$ we have

$$S_{t+1} = \mathrm{Poi}\left(\sum_{l=1}^{S_t} X_t^{(l)} \right), \tag{22}$$

where $\left(X_t^{(l)}\right)_l$ are i.i.d. copies of $\frac{a+b}{2}\Phi^{(1)}\phi^*$, where ϕ^* follows law ν^*.

Note that in the *ordinary* Stochastic Block Model (i.e., when all vertices have unit weight), the argument of the Poisson random variables in (22) is deterministic, contrary to the general case under consideration here. Using (17) recursively in conjunction with (18), it follows that

$$\mathbb{E}\left[S_t|\phi_o\right] = \frac{\Phi^{(1)}\phi_o}{\Phi^{(2)}}\rho^t, \quad \forall t \geq 1.$$

In the following lemma we show that deviations from this average are small. In fact, there exists a constant C such that for each $t \geq 0$, S_t is asymptotically stochastically dominated by an Exponential random variable with mean $C\rho^t$. An important ingredient in the proof below is Hoeffding's inequality, which we use to derive a concentration result for the parameter of the Poisson variable in (22).

▶ **Lemma 10** (Degree-Corrected Extension of Lemma 23 in [2]). *Assume $S_0 = 1$. There exist $c, c' > 0$ such that for all $s \geq 0$,*

$$\mathbb{P}\left(\forall k \geq 1, S_k \leq s\rho^k\right) \geq 1 - c'e^{-cs}.$$

From Theorem 7 and Corollary 8, we know that the different components (expressed in the basis of eigenvectors of M) grow exponentially with rate ρ, respectively μ_2. We now quantify the error. Recall Ψ_t from (21).

5.2.1 The case $\mu_2^2 > \rho$

▶ **Theorem 11** (Degree-Corrected Extension of Theorem 24 in [2]). *Assume that $\mu_2^2 > \rho$. Let $\beta > 0$, $Z_0 = \delta_x$ and $\phi_o = \psi_o$ be fixed. There exists $C = C(x, \beta) > 0$ such that with probability at least $1 - n^{-\beta}$, for all $k \in \{1, 2\}$, all $0 \leq s < t \leq C_{min}\log(n)$, with $0 \leq s < t$,*

$$|\langle g_k, Z_s\rangle - \mu_k^{s-t}\langle g_k, Z_t\rangle| \leq C(s+1)\rho^{s/2}(\log n)^{3/2},$$

and,

$$|\langle g_k, \Psi_s\rangle - \mu_k^{s-t}\langle g_k, \Psi_t\rangle| \leq C\rho^{s/2}(\log n)^{5/2}.$$

5.2.2 The case $\mu_2^2 \leq \rho$

▶ **Theorem 12.** *Assume that $\mu_2^2 \leq \rho$. Let $\beta > 0$, $Z_0 = \delta_x$ and $\phi_o = \psi_o$ be fixed. There exists $C = C(x, \beta) > 0$ such that with probability at least $1 - n^{-\beta}$, for all $t \geq 1$,*

$$|\langle g_2, \Psi_t\rangle| \leq Ct^2\rho^{t/2}(\log n)^2,$$

and,

$$\mathbb{E}\left[|\langle g_2, \Psi_t\rangle|^2\right] \leq Ct^3\rho^t.$$

5.3 $B^\ell B^{*\ell}\check{\chi}_k$ on trees: a cross generation functional

Recall our claim that $B^\ell B^{*\ell}\check{\chi}_k$ are asymptotically aligned with the eigenvectors of B. In the DC-SBM, the local-neighbourhood of a vertex has with high probability a tree-like structure described by the branching process above. In this section we analyse $B^\ell B^{*\ell}\check{\chi}_k$ on trees.

To this end we define a cross-generational functional slightly different from its analogue in [2] due to the presence of weights:

$$Q_{k,\ell} = \sum_{(u_0,\dots,u_{2\ell+1})\in\mathcal{P}_{2\ell+1}} g_k(\sigma(u_{2\ell+1}))\phi_{u_{2\ell+1}}, \tag{23}$$

where $\mathcal{P}_{2\ell+1}$ is the set of paths $(u_0,\dots,u_{2\ell+1})$ (of length $2\ell+1$) in the tree starting from $u_0 = o$ with both $(u_0,\dots,u_\ell)$ and $(u_\ell,\dots,u_{2\ell+1})$ non-backtracking and $u_{\ell-1} = u_{\ell+1}$. Note that these paths thus make a back-track exactly once at step $\ell+1$.

Explicitly, we have

$$Q_{1,\ell} = \sum_{(u_0,\dots,u_{2\ell+1})\in\mathcal{P}_{2\ell+1}} \frac{1}{\sqrt{2}}\phi_{u_{2\ell+1}}, \tag{24}$$

and,

$$Q_{2,\ell} = \sum_{(u_0,\dots,u_{2\ell+1})\in\mathcal{P}_{2\ell+1}} \frac{1}{\sqrt{2}}\sigma(u_{2\ell+1})\phi_{u_{2\ell+1}}. \tag{25}$$

Consider a tree $\mathcal{T}'$ and a leaf e_1 on it that has unique neighbour, say, o. Then, if e is the oriented edges from e_1 to o and if $B_{\mathcal{T}'}$ denotes the non-backtracking matrix defined on $\mathcal{T}'$,

$$\left(B^{\ell}_{\mathcal{T}'}B^{*\ell}_{\mathcal{T}'}\check{\chi}_k\right)(e) = Q_{k,\ell} + g_k(\sigma(e_1))\phi_{e_1}\|Z_\ell\|_1, \tag{26}$$

where $Q_{k,\ell}$ and Z_ℓ are defined on the tree $\mathcal{T}$ with root o obtained after removing vertex e_1 from $\mathcal{T}'$.

In the sequel we analyse $Q_{k,\ell}$ on the branching process defined above, starting with a single particle, the root o. Let V indicate the particles of the random tree. Denote the spin of a particle $v\in V$ by $\sigma_v\in\{+,-\}$ and its weight by $\phi_v\in S$.

For $t\geq 0$, let Y_t^v denote the set of particles, *including* their spins and weights, of generation t from v in the subtree of particles with common ancestor $v\in V$. Let $Z_t^v = (Z_t^{v,+}, Z_t^{v,-})$ denote the number of $\pm$ vertices in generation t; i.e., $Z_t^{v,\pm} = \sum_{u\in Y_t^v} 1_{\sigma(u)=\pm}$. Finally, let $\Psi_t^v = (\Psi_t^{v,+}, \Psi_t^{v,-})$, with $\Psi_t^{v,\pm} = \sum_{u\in Y_t^v} 1_{\sigma(u)=\pm}\phi_u$.

We rewrite $Q_{k,\ell}$ into a more manageable form: First observe that every path in $\mathcal{P}_{2\ell+1}$, after reaching $u_{\ell+1}$, climbs back to a depth t from which it then again moves down the tree (that is, in the direction away from the root). Let us call the vertex at level t (to which the path climbs back before descending again) u. Then, (if $t\neq 0$) there are two children of u, say v and w such that w lies on the path between u and $u_{\ell+1}$ and v is in between u and $u_{2\ell+1}$. For such fixed v and w in Y_1^u, only the children $u_{2\ell+1}\in Y_t^v$ determine the contribution of a path to (23), regardless of the choice of $u_{\ell+1}\in Y_{\ell-t-1}^w$. Hence, for such fixed u and $v,w\in Y_1^u$ and $u_{2\ell+1}$, there are $|Y_{\ell-t-1}^w| = S_{\ell-t-1}^w$ paths giving the same contribution to (23):

$$Q_{k,\ell} = \sum_{t=0}^{\ell-1}\sum_{u\in Y_t^o} L_{k,\ell}^u, \tag{27}$$

where, for $|u| = t\geq 0$,

$$L_{k,\ell}^u = \sum_{w\in Y_1^u} S_{\ell-t-1}^w \left(\sum_{v\in Y_1^u\setminus\{w\}} \langle g_k, \Psi_t^v\rangle\right). \tag{28}$$

The following theorem is an extension of Theorem 25 in [2]. The important observation is that, again, for $Z_0 = \delta_\tau$ fixed, $\left(Q_{2,\ell}/\mu_2^{2\ell}\right)_\ell$ converges to a random variable with mean a

constant times τ, that is, the spin of the root. Its proof uses both martingale theorems stated above. We use the second martingale statement, which is not present in the ordinary SBM, to bound the variance of $Q_{k,\ell}$:

▶ **Theorem 13** (Degree-Corrected Extension of Theorem 25 in [2]). *Assume that $\mu_2^2 > \rho$. Let $Z_0 = \delta_x$ and $\phi_o = \psi_o$ be fixed. For $k \in \{1,2\}$, $\left(Q_{k,\ell}/\mu_k^{2\ell}\right)_\ell$ converges in L^2 as ℓ tends to infinity to a random variable with mean $\frac{\Phi^{(3)}}{\Phi^{(2)}} \frac{\rho}{\mu_k^2 - \rho} \mu_{k,\psi_o} g_k(x)$. Further, the L^2-convergence takes place uniformly for all ψ_o.*

5.3.1 The case $\mu_2^2 \leq \rho$

▶ **Theorem 14.** *Assume that $\mu_2^2 \leq \rho$. Let $Z_0 = \delta_x$ and $\phi_o = \psi_o$ be fixed. There exists a constant C such that $\mathbb{E}\left[Q_{2,\ell}^2\right] \leq C\rho^{2\ell}\ell^5$.*

5.4 Orthogonality: Decorrelation in branching process

Again, as in [2], $Q_{1,\ell}$ and $Q_{2,\ell}$ are uncorrelated when defined on the branching process above. The proof presented here is simpler than the corresponding one in [2] and uses that for the two communities-case, $Q_{1,\ell}$ and $Q_{2,\ell}$ are *explicitly* known.

The orthogonality of the candidate eigenvectors (i.e., $(iii) - (v)$ in Proposition 4) follows from this fact, see Proposition 24 $(ii), (iii)$ and Proposition 25 (ii) below.

▶ **Theorem 15** (Degree-Corrected Extension of 28 in [2]). *Assume that the spin σ_o of the root is drawn uniformly from $\{+,-\}$. Then for any $\ell \geq 0$,*

$$\mathbb{E}\left[Q_{1,\ell} Q_{2,\ell} | \mathcal{T}\right] = 0.$$

6 Coupling of local neighbourhood

The proofs of the statements in this section can be found in Appendix B in the detailed version of the underlying article (Arxiv:1609.02487).

6.1 Coupling

Here we establish the connection between neighbourhoods in the DC-SBM and the branching process in Section 5. We established this coupling in an earlier paper [6] using an exploration process that we repeat below. Compared to the ordinary SBM, vertices are now weighted, so that two facts need to be verified: At each step of the exploration process, unexplored vertices have a weight drawn from a distribution close in total variation distance to ν. Detected vertices on their turn follow a law close to ν^*.

We distinguish between two different concepts of neighbourhood: the classical neighbourhood that is rooted at a vertex and another neighbourhood that starts with an edge. For the latter, we need the following concept of *oriented* distance $\vec{d}$, which for $e, f \in \vec{E}(V)$ is defined as

$$\vec{d}(e,f) = \min_\gamma \ell(\gamma)$$

where the minimum is taken over all self-avoiding paths $\gamma = (\gamma_0, \gamma_1, \cdots, \gamma_{\ell+1})$ in G such that $(\gamma_0, \gamma_1) = e$, $(\gamma_\ell, \gamma_{\ell+1}) = f$ and for all $1 \leq k \leq \ell + 1$, $\{\gamma_k, \gamma_{k+1}\} \in E$. and where for such a path γ, $\ell(\gamma) = \ell$. Note that $\vec{d}(e,f) = \vec{d}(f^{-1}, e^{-1})$, i.e., $\vec{d}$ is not symmetric.

We introduce the vector $Y_t(e) = (Y_t(e)(i))_{i\in\{+,-\}}$ where, for $i \in \{+,-\}$,

$$Y_t(e)(i) = \left|\left\{f \in \vec{E} : \vec{d}(e,f) = t, \sigma(f_2) = i\right\}\right|, \tag{29}$$

we denote the number of vertices at oriented distance t from e by

$$S_t(e) = \|Y_t(e)\|_1 = \left|\left\{f \in \vec{E} : \vec{d}(e,f) = t\right\}\right|,$$

and we define vector $\Psi_t(e) = (\Psi_t(e)(i))_{i\in\{+,-\}}$ where, for $i \in \{+,-\}$,

$$\Psi_t(e)(i) = \sum_{f\in\vec{E}:\vec{d}(e,f)=t} 1_{\sigma(f_2)=i}\phi_{f_2}. \tag{30}$$

We denote the classical neighbourhood of radius r rooted at vertex v by $(G,v)_r$ and the neighbourhood around oriented edge $e = (e_1,e_2)$ by $(G,e)_r$. With the definitions above, we then have, $(G,e)_r = (G',e_2)_r$, where G' is the graph G with edge $\{e_1,e_2\}$ removed. In particular,

$$S_t(e) = S'_t(e_2),$$

where S'_t is S_t defined on G'.

The two branching processes that describe the neighbourhoods are almost identical, the only difference lies in the weight of the root: In the classical branching processes, the weight is drawn according to distribution ν. In the branching process starting at an edge oriented towards, say, o, the root o has weight governed by ν^*. See Proposition 16 below.

As a corollary we obtain an analogue of Theorem 11 for local neighbourhoods: the components of $\Psi_t(e)$ grow exponentially, see Corollary 17.

We bound the growth of S_t in Lemma 18. We use a coupling argument to show that the weights of the unexplored vertices and selected vertices are stochastically dominated by variables following law ν, respectively ν^*. This argument is not needed in the ordinary SBM.

Following [17], we need to verify that certain problematic structures, namely *tangles*, are excluded with high probability. We say that a graph H is tangle-free if all its $\ell-$ neighbourhoods contain at most one cycle. If there is at least one $\ell-$ neighbourhood in H that contains more than one cycle, we call H tangled. Note that in the sequel we shall often suppress the dependence on ℓ and simply call a graph tangle-free or tangled; the ℓ dependence is then tacitly assumed.

Following standard arguments we establish in Lemma 19 that the graph is with high probability $\log(n)$-tangle free.

We prepare by recalling the exploration process in [6] starting at a vertex:

At time $m = 0$, choose a vertex ρ in $V(G)$, where G is an instant of the DC-SBM. Initially, it is the only active vertex: $\mathcal{A}(0) = \{\rho\}$. All other vertices are neutral at start: $\mathcal{U}(0) = V(G) \setminus \{\rho\}$. No vertex has been explored yet: $\mathcal{E}(0) = \emptyset$.

At each time $m \geq 0$ we arbitrarily pick an active vertex u in $\mathcal{A}(m)$ that has shortest distance to ρ, and explore all its edges in $\{uv : v \in \mathcal{U}(m)\}$: if $uv \in E(G)$ for $v \in \mathcal{U}(m)$, then we set v active in step $m+1$, otherwise it remains neutral.

At the end of step m, we designate u to be explorated.

Thus,

$$\mathcal{E}(m+1) = \mathcal{E}(m) \cup \{u\},$$

$$\mathcal{A}(m+1) = (\mathcal{A}(m) \setminus \{u\}) \cup (\mathcal{N}(u) \cap \mathcal{U}(m)),$$

and,

$$\mathcal{U}(m+1) = \mathcal{U}(m) \setminus \mathcal{N}(u).$$

▶ **Proposition 16** (Degree-Corrected Extension of Proposition 31 in [2]). *Let $\ell = C\log_\rho(n)$, with $C < C_{coupling}$. Let $\rho \in V$ and $e = (e_1, e_2) \in \vec{E}$. Let (T, o) be the branching process with root o defined in Section 5, where the root has spin $\sigma(v)$ and weight governed by ν. Similarly, Let (T', o) be that same branching process, when the root has spin $\sigma(e_2)$ and weight governed by ν^*. Then, the total variation distance between the law of $(G, v)_\ell$ and $(T, o)_\ell$ goes to zero as $1 - n^{-\left(\frac{\gamma}{2} \wedge \frac{1}{40}\right)}$. The same is true for the difference between the law of $(G, e)_\ell$ and (T', o).*

▶ Remark. Note that with the event $(G, v)_\ell = (T, o)_\ell$, we mean that the graph and tree are equal, **including** their **spins and weights**. See [6] for more details.

▶ **Corollary 17** (Degree-Corrected Extension of Corollary 32 in [2]). *Assume $\mu_2^2 > \rho$. Let $\ell = C\log_\rho n$ with $0 < C < C_{coupling}$. For $e \in \vec{E}(V)$, we define the event $\mathcal{E}(e)$ that for all $0 \le t < \ell$ and $k \in \{1, 2\}$: $|\langle g_k, \Psi_t(e)\rangle - \mu_k^{t-\ell}\langle g_k, \Psi_\ell(e)\rangle| \le (\log n)^3 \rho^{t/2}$. Then, with high probability, the number of edges $e \in \vec{E}$ such that $\mathcal{E}(e)$ does not hold is at most $\log(n)\ n^{1-(\frac{\gamma}{2} \wedge \frac{1}{40})}$.*

▶ **Lemma 18** (Degree-Corrected Extension of Lemma 29 in [2]). *There exist $c, c' > 0$ such that for all $s \ge 0$ and for any $w \in [n] \cup \vec{E}(V)$,*

$$\mathbb{P}\left(\forall t \ge 0 : S_t(w) \le s\vec{\rho}_n^t\right) \ge 1 - ce^{-c's}.$$

Consequently, for any $p \ge 1$, there exists $c'' > 0$ such that

$$\mathbb{E}\left[\max_{v \in [n], t \ge 0} \left(\frac{S_t(v)}{\vec{\rho}_n^t}\right)^p\right] \le c''(\log n)^p.$$

▶ **Lemma 19** (Degree-Corrected Extension of Lemma 30 in [2]). *Let $\ell = C\log_\rho(n)$, with $0 < C < C_{coupling}$. Then, w.h.p., at most $\rho^{2\ell}\log(n)$ vertices have a cycle in their ℓ - neighbourhood. Further, w.h.p., the graph is ℓ - tangle-free.*

6.2 Geometric growth

Here we show that for $k \in \{1, 2\}$, $\langle B^\ell \chi_k, \delta_e\rangle$ grows nearly geometrically in t with rate μ_k. Corollary 21 then establishes a bound for $r \le \ell$ on $\sup_{\langle B^\ell\chi_k, x\rangle = 0, \|x\|=1} \|\langle B^r\chi_k, x\rangle\|$ crucial for the norm bounds in Section 9.

▶ **Proposition 20** (Degree-Corrected Extension of Proposition 33 in [2]). *Assume $\mu_2^2 > \rho$. Let $\ell = C\log_\rho(n)$, with $0 < C < C_{coupling} \wedge \left(\frac{1}{2} - \left(\frac{\gamma}{4} \wedge \frac{1}{80}\right)\right) = C_{coupling}$. For $e \in \vec{E}(V)$, let $\vec{E}_\ell$ be the set of oriented edges such that either $(G, e_2)_\ell$ is not a tree or the event $\mathcal{E}(e)$ (defined in Corollary 17) does not hold. Then, w.h.p. for $k \in \{1, 2\}$:*

(i) $|\vec{E}_\ell| \ll (\log n)^2 n^{1-\frac{\gamma}{2} \wedge \frac{1}{40}}$,

(ii) *for all $e \in \vec{E} \setminus \vec{E}_\ell$, $0 \le r \le \ell$,*

$$|\langle B^r\chi_k, \delta_e\rangle - \mu_k^{r-\ell}\langle B^\ell\chi_k, \delta_e\rangle| \le (\log n)^4 \rho^{r/2},$$

(iii) *for all $e \in \vec{E}_\ell$, $0 \le r \le \ell$,*

$$|\langle B^r\chi_k, \delta_e\rangle| \le (\log n)^2 \rho^r.$$

▶ **Corollary 21** (Degree-Corrected Extension of Corollary 34 in [2]). *Let $\ell = C\log_\rho(n)$, with $0 < C < C_{coupling} \wedge \left(1 - \frac{\gamma}{2} \wedge \frac{1}{40}\right) \wedge \left(\frac{\gamma}{4} \wedge \frac{1}{80}\right) = C_{coupling}$. W.h.p. for any $0 \le r \le \ell - 1$ and $k \in \{1, 2\}$:*

$$\sup_{\langle B^\ell\chi_k, x\rangle = 0, \|x\|=1} \|\langle B^r\chi_k, x\rangle\| \le (\log n)^5 n^{1/2} \rho^{r/2}.$$

7 A weak law of large numbers for local functionals on the DC-SBM

The proofs of the statements in this section can be found in Appendix C in the detailed version of the underlying article (Arxiv:1609.02487).

Here we show that a weak law of large numbers applies for local functionals defined on *weighted coloured* random graphs generated according to the DC-SBM.

By a *weighted coloured* graph we mean a graph $G = (V, E)$ together with maps $\sigma : V \to \{+, -\}$ and $\phi : V \to [\phi_{\min}, \phi_{\max}]$. For $v \in V$, we identify $\sigma(v)$ as the spin of v and $\phi(v)$ as its weight. We denote by $\mathcal{G}^*$ the set of *rooted weighted coloured* graphs. We denote an element of $\mathcal{G}^*$ by (G, o): $G = (V, E)$ is then a weighted coloured graph and $o \in V$ is some distinguished vertex. A function $\tau : \mathcal{G}^* \to \mathbb{R}$ is said to be $\ell-$ local if $\tau(G, o)$ depends only on $(G, o)_\ell$.

To derive the claimed weak law when G is drawn according to the DC-SBM, we prepare with a variance bound for $\sum_{v=1}^n \tau(G, v)$, see Proposition 22. The bound follows from the law of total variance,

$$\begin{aligned}\operatorname{Var}\left(\sum_{v=1}^n \tau(G,v)\right) &= \mathbb{E}\left[\operatorname{Var}\left(\sum_{v=1}^n \tau(G,v)\middle|\, \phi_1,\dots,\phi_n\right)\right] \\ &\quad + \operatorname{Var}\left(\mathbb{E}\left[\sum_{v=1}^n \tau(G,v)\middle|\, \phi_1,\dots,\phi_n\right]\right),\end{aligned}$$

together with an application of Efron-Stein's inequality to both terms on the right. Note that $\mathbb{E}\left[\sum_{v=1}^n \tau(G, v)|\, \phi_1, \dots, \phi_n\right]$ is a constant in the *ordinary* SBM, whereas here it needs a careful analysis.

The sample average $\frac{1}{n}\sum_{v=1}^n \tau(G, v)$ concentrates then around $\mathbb{E}[\tau(T, o)]$, where (T, o) is the branching process from Section 5, with root o having spin drawn uniformly from $\{+, -\}$ and weight governed by ν, see Proposition 23. The coupling, and in particular the matching of the weights, plays an important role in its proof.

In the next section we apply the latter proposition to some specific functionals.

▶ **Proposition 22** (Degree-Corrected Extension of Proposition 35 in [2]). *Let G be drawn according to the DC-SBM. There exists $c > 0$ such that if $\tau, \varphi : \mathcal{G}^* \to \mathbb{R}$ are ℓ-local, $|\tau(G, o)| \le \varphi(G, o)$ and φ is non-decreasing by the addition of edges, then*

$$\operatorname{Var}\left(\sum_{v=1}^n \tau(G,v)\right) \le cn\rho^{2\ell}\left(\mathbb{E}\left[\max_{v\in[n]} \varphi^4(G,v)\right]\right)^{1/2}.$$

▶ **Proposition 23** (Degree-Corrected Extension of Proposition 36 in [2]). *Let G be drawn according to the DC-SBM. Let (T, o) be the branching process from Section 5, with root o having spin drawn uniformly from $\{+, -\}$ and weight governed by ν. Let $\ell = C\log_\rho(n)$, with $C < C_{coupling}$. There exists $c > 0$ such that if $\tau, \varphi : \mathcal{G}^* \to \mathbb{R}$ are ℓ-local, $|\tau(G, o)| \le \varphi(G, o)$ and φ is non-decreasing by the addition of edges, then*

$$\begin{aligned}&\mathbb{E}\left[\left|\frac{1}{n}\sum_{v=1}^n \tau(G,v) - \mathbb{E}[\tau(T,o)]\right|\right] \\ &\le c_2 n^{-\left(\frac{\gamma}{2}\wedge\frac{1}{40}\right)}\left(\mathbb{E}\left[\max_{v\in[n]}\varphi^4(G,v)\right]^{1/4} \vee \mathbb{E}\left[\varphi^2(T,o)\right]^{1/2}\right) + \mathcal{O}(n^{-\gamma})\end{aligned} \tag{31}$$

7.1 Application with some specific local functionals

Here we consider $\langle B^\ell\chi_1, B^\ell\chi_2\rangle$, $\langle B^{2\ell}\chi_k, B^\ell\chi_j\rangle$, and $\langle B^\ell B^{*\ell}\chi_1, B^\ell B^{*\ell}\chi_2\rangle$, quantities occurring in Proposition 4.

Explicitly, $B^\ell \chi_k(e) = \sum_f B^\ell_{ef} g_k(\sigma(f_2))\phi_{f_2}$, where we recall that B^ℓ_{ef} is the number of non-backtracking walks from e to f. Now, if the oriented $\ell-$ neighbourhood of e is a *tree*, then $B^\ell \chi_k(e) = \langle g_k, \Psi_\ell(e)\rangle$. With this intuition in mind, we analyse likewise expressions in Proposition 24 below.

Inspired by (26), which expresses $B^\ell B^{*\ell}\chi_k$ on *trees* in terms of the operator $Q_{k,\ell}$, we extend the latter to an operator defined on general graphs. First, for $e \in \vec{E}(V)$ and $t \geq 0$, set $\mathcal{Y}_t(e) = \{f \in \vec{E} : \vec{d}(e,f) = t\}$. Then, for $k \in \{1,2\}$, we set

$$P_{k,\ell}(e) = \sum_{t=0}^{\ell-1} \sum_{f \in \mathcal{Y}_t(e)} L_k(f), \tag{32}$$

with

$$L_k(f) = \sum_{(g,h)\in \mathcal{Y}_1(f)\setminus \mathcal{Y}_t(e); g\neq h} \langle g_k, \tilde{\Psi}_t(g)\rangle \tilde{S}_{\ell-t-1}(h),$$

where $\tilde{\Psi}_t(g)$, $\tilde{S}_{\ell-t-1}(h) = \|\tilde{Y}_{\ell-t-1}(h)\|_1$ are the variables $\Psi_t(g)$, respectively $S_{\ell-t-1}(h)$, defined on the graph G where all edges in $(G,e_2)_t$ have been removed. Note that, if $(G,e)_{2\ell}$ is a tree, then $\tilde{\Psi}_s(g) = \Psi_s(g)$ for $s \leq 2\ell - t$. Compare $P_{k,\ell}$ to $Q_{k,\ell}$ in (23) and $L_k(f)$ to $L^u_{k,\ell}$ in (28).

Finally, define

$$S_{k,\ell}(e) = S_\ell(e) g_k(\sigma(e_1))\phi_{e_1}. \tag{33}$$

We then have an extension of (26), when $(G,e_2)_{2\ell}$ is a tree:

$$B^\ell B^{*\ell}\check{\chi}_k(e) = P_{k,\ell}(e) + S_{k,\ell}(e). \tag{34}$$

We analyse (34) in Proposition 25 below.

7.1.1 The case $\mu_2^2 > \rho$

▶ **Proposition 24** (Degree-Corrected Extension of Proposition 37 in [2]). *Assume that $\mu_2^2 > \rho$. Let $\ell = C\log_\rho n$ with $0 < C < C_{coupling}$.*

(i) *For any $k \in \{1,2\}$, there exists $c'_k > 0$ such that, in probability,*

$$\frac{1}{n}\sum_{e\in\vec{E}} \frac{\langle g_k, \Psi_\ell(e)\rangle^2}{\mu_k^{2\ell}} \to c'_k.$$

(ii) *For any $k \in \{1,2\}$, there exists $c''_k > 0$ such that, in probability,*

$$\frac{1}{n}\sum_{e\in\vec{E}} \frac{\langle g_k, Y_\ell(e)\rangle^2}{\mu_k^{2\ell}} \to c''_k.$$

(iii)

$$\mathbb{E}\left[\left|\frac{1}{n}\sum_{e\in\vec{E}} \langle g_1, \Psi_\ell(e)\rangle\langle g_2, \Psi_\ell(e)\rangle\right|\right] \leq (\log n)^3 n^{2C-\left(\frac{\gamma}{2}\wedge\frac{1}{40}\right)} + n^{-\gamma}.$$

(iv) *For any $k \neq j \in \{1,2\}$,*

$$\mathbb{E}\left[\left|\frac{1}{n}\sum_{e\in\vec{E}} \langle g_k, \Psi_{2\ell}(e)\rangle\langle g_j, \Psi_\ell(e)\rangle\right|\right] \leq (\log n)^3 n^{3C-\left(\frac{\gamma}{2}\wedge\frac{1}{40}\right)} + n^{-\gamma}.$$

(v) *For any $k \in \{1,2\}$, in probability*

$$\frac{1}{n}\sum_{e\in\vec{E}} \frac{\langle g_k, \Psi_{2\ell}(e)\rangle\langle g_k, \Psi_\ell(e)\rangle}{\mu_k^{3\ell}} \to c_k'''.$$

▶ **Proposition 25** (Degree-Corrected Extension of Proposition 38 in [2]). *Assume that $\mu_2^2 > \rho$. Let $\ell = C\log_\rho n$ with $C < C_{coupling}$.*

(i) *For any $k \in \{1,2\}$, there exists $c_k'''' > 0$ such that in probability*

$$\frac{1}{n}\sum_{e\in\vec{E}} \frac{P_{k,\ell}^2(e)}{\mu_k^{4\ell}} \to c_k''''.$$

(ii)

$$\mathbb{E}\left[\left|\frac{1}{n}\sum_{e\in\vec{E}}(P_{1,\ell}(e)+S_{1,\ell}(e))(P_{2,\ell}(e)+S_{2,\ell}(e))\right|\right] \le (\log n)^8 n^{4C-\left(\frac{\gamma}{2}\wedge\frac{1}{40}\right)}$$

7.1.2 The case $\mu_2^2 \le \rho$

Most of the above claims continue to hold if $\mu_2^2 \le \rho$. We treat the exceptions here.

▶ **Proposition 26.** *Assume that $\mu_2^2 \le \rho$. Let $\ell = C\log_\rho n$ with $0 < C < C_{coupling}$. There exists some $c > 0$, such that w.h.p.,*

$$\frac{1}{n}\sum_{e\in\vec{E}} \frac{\langle g_2, \Psi_\ell(e)\rangle^2}{\rho^\ell} \ge c.$$

▶ **Proposition 27.** *Assume that $\mu_2^2 \le \rho$. Let $\ell = C\log_\rho n$ with $C < C_{coupling}$. There exists $c > 0$ such that w.h.p.,*

$$\frac{1}{n}\sum_{e\in\vec{E}} \frac{P_{2,\ell}^2(e)}{\rho^{2\ell}\log^5(n)} \le c.$$

8 Proof op Propositions 4 and 6

We introduce for $k \in \{1,2\}$ the vector $N_{k,\ell}$, defined on $e \in \vec{E}$ as

$$N_{k,\ell}(e) = \langle g_k, \Psi_\ell(e)\rangle.$$

If $(G, e_2)_\ell$ is a tree, then

$$N_{k,\ell}(e) = \langle B^\ell \chi_k, \delta_e\rangle,$$

and we have a similar expression for $B^\ell B^{*\ell}\check{\chi}_k$ in (34). Now, at most $\rho^{2\ell}\log(n)$ vertices have a cycle in their ℓ-neighbourhood (see Lemma 19). Therefore:

▶ **Lemma 28** (Degree-Corrected Extension of Lemma 39 in [2]). *Let $\ell = C\log_\rho n$ with $0 < C < C_{min}$. Then, w.h.p. $\|B^\ell\chi_k - N_{k,\ell}\| = O\left((\log n)^{5/2}\rho^{2\ell}\right) = o\left(\rho^{\ell/2}\sqrt{n}\right)$, $\|B^\ell B^{*\ell}\check{\chi}_k - P_{k,\ell} - S_{k,\ell}\| = O((\log n)^4\rho^{4\ell})$ and $\|B^\ell B^{*\ell}\check{\chi}_k - P_{k,\ell}\| = O(\rho^\ell\sqrt{n})$.*

Proof. The proof of Lemma 39 in [2] can be easily adapted to the current setting. The key idea is pointed out above. It thus remains to bound $|(B^\ell \chi_k - N_{k,\ell})(e)|$ and $|(B^\ell B^{*\ell}\check{\chi}_k - P_{k,\ell})(e)|$ on edges e for which $(G, e_2)_\ell$ is not a tree. For this, use that with high probability the graph is 2ℓ-tangle free so that there are at most two non-backtracking paths between e and any edge at distance ℓ. ◀

We can thus in our calculations replace $B^\ell \chi_k$ by $N_{k,\ell}$ and $B^\ell B^{*\ell}\check{\chi}_k$ by $P_{k,\ell}$. From Propositions 24 and 25, Proposition 4 then follows:

Proof of Proposition 4. This proof follows the corresponding proof in [2]. We give the key observations: (i) From Proposition 24 (i), $\|N_{k,\ell}\| \sim \sqrt{n}\mu_k^\ell$ and from Proposition 25 (i), $\|P_{k,\ell}\| \sim \sqrt{n}\mu_k^{2\ell}$.

(ii) From Proposition 24 (v), $|\langle N_{k,\ell}, N_{k,2\ell}\rangle| \sim n\mu_k^{3\ell}$.

(iii) From Proposition 24 (iii), $|\langle N_{1,\ell}, N_{2,\ell}\rangle| \sim (\log n)^3 n^{3C-\left(\frac{\gamma}{2}\wedge\frac{1}{40}\right)}$.

(iv) From Proposition 24 (iv), $|\langle N_{k,2\ell}, N_{j,\ell}\rangle| \sim (\log n)^3 n^{4C-\left(\frac{\gamma}{2}\wedge\frac{1}{40}\right)}$.

(v) From Proposition 25 (ii), $|\langle P_{1,\ell} + S_{1,\ell}, P_{2,\ell} + S_{2,\ell}\rangle| \sim (\log n)^8 n^{5C-\left(\frac{\gamma}{2}\wedge\frac{1}{40}\right)}$. ◀

Proposition 6 follows similarly from the case $\mu_2^2 \le \rho$ treated in Section 7.1:

Proof of Proposition 6. This follows from Propositions 26 and 27 in conjunction with Lemma 28. ◀

9 Norm of non-backtracking matrices

The proofs of the statements in this section can be found in Appendix D in the detailed version of the underlying article (Arxiv:1609.02487).

In this section the product over an empty set is defined to be one.

It is convenient to extend matrix B and vector χ_k to the set of directed edges on the *complete* graph, $\vec{E}_K(V) = \{(u,v) : u \neq v \in V\}$: For $e, f \in \vec{E}_K(V)$, B_{ef} is then extended to

$$B_{ef} = A_e A_f 1_{e_2=f_1} 1_{e_1 \neq f_2}, \tag{35}$$

where A is the adjacency matrix. For each $e \in \vec{E}_K(V)$ we set $\chi_k(e) = g_k(\sigma(e_2))\phi_{e_2}$.

For integer $k \ge 1$, $e, f \in \vec{E}_K(V)$, we let Γ_{ef}^k be the set of non-backtracking walks $\gamma = (\gamma_0, \dots, \gamma_k)$ of length k from $(\gamma_0, \gamma_1) = e$ to $(\gamma_{k-1}, \gamma_k) = f$ on the *complete* graph with vertex set V.

By induction it follows that

$$(B^k)_{ef} = \sum_{\gamma \in \Gamma_{ef}^{k+1}} \prod_{s=0}^{k} A_{\gamma_s \gamma_{s+1}}. \tag{36}$$

Indeed, note that $\prod_{s=0}^k A_{\gamma_s\gamma_{s+1}}$ is one when γ is a path in G and zero otherwise.

To each walk $\gamma = (\gamma_0, \dots, \gamma_k)$, we associate the graph $G(\gamma) = (V(\gamma), E(\gamma))$, with the set of vertices $V(\gamma) = \{\gamma_i, 0 \le i \le k\}$ and the set of edges $E(\gamma) = \{\{\gamma_i, \gamma_{i+1}\}, 0 \le i \le k-1\}$.

From Lemma 19, the graphs following the DC-SBM are tangle-free with high probability. Hence, it makes sense to consider the subset $F_{ef}^{k+1} \subset \Gamma_{ef}^{k+1}$ of tangle-free non-backtracking walks on the *complete* graph. Indeed, if G is tangle-free, we need only consider the tangle-free paths in the summation (36):

$$(B^{(k)})_{ef} = \sum_{\gamma \in F_{ef}^{k+1}} \prod_{s=0}^{k} A_{\gamma_s \gamma_{s+1}}, \tag{37}$$

and $B^k = B^{(k)}$ for $1 \leq k \leq \ell$.

Define for $u \neq v$ the *centred* random variable

$$\underline{A}_{uv} = A_{uv} - \frac{\phi_u \phi_v}{n} W_{\sigma_u \sigma_v}, \tag{38}$$

where

$$W = \begin{pmatrix} a & b \\ b & a \end{pmatrix}.$$

Compare this to the SBM *without* degree-corrections in Section 10.1 **of [2]:** $\phi_u = 1$ **for all** u **in the latter model.**

Using $\underline{A}$ we shall attempt to center B^k when the underlying graph G is tangle-free through considering

$$\Delta_{ef}^{(k)} = \sum_{\gamma \in F_{ef}^{k+1}} \prod_{s=0}^{k} \underline{A}_{\gamma_s \gamma_{s+1}}. \tag{39}$$

Further, we set

$$\Delta_{ef}^{(0)} = 1_{e=f} \underline{A}_e \quad \text{and} \quad B_{ef}^{(0)} = 1_{e=f} A_e. \tag{40}$$

To decompose (37), following a decomposition that appeared first in [16], we use

$$\prod_{s=0}^{\ell} x_s = \prod_{s=0}^{\ell} y_s + \sum_{t=0}^{\ell} \prod_{s=0}^{t-1} y_s (x_t - y_t) \prod_{s=t+1}^{\ell} x_s,$$

with $x_s = A_{\gamma_s \gamma_{s+1}}$ and $y_s = \underline{A}_{\gamma_s \gamma_{s+1}}$ on a path $\gamma \in F_{ef}^{k+1}$:

$$\prod_{s=0}^{\ell} A_{\gamma_s \gamma_{s+1}} = \prod_{s=0}^{\ell} \underline{A}_{\gamma_s \gamma_{s+1}} + \sum_{t=0}^{\ell} \prod_{s=0}^{t-1} \underline{A}_{\gamma_s \gamma_{s+1}} \left(\frac{\phi_{\gamma_t} \phi_{\gamma_{t+1}}}{n} W_{\sigma_{\gamma_t} \sigma_{\gamma_{t+1}}} \right) \prod_{s=t+1}^{\ell} A_{\gamma_s \gamma_{s+1}}.$$

Summing over all $\gamma \in F_{ef}^{\ell+1}$ then gives

$$\begin{aligned} B_{ef}^{(\ell)} &= \sum_{\gamma \in F_{ef}^{\ell+1}} \prod_{s=0}^{\ell} \underline{A}_{\gamma_s \gamma_{s+1}} \\ &\quad + \sum_{t=0}^{\ell} \sum_{\gamma \in F_{ef}^{\ell+1}} \prod_{s=0}^{t-1} \underline{A}_{\gamma_s \gamma_{s+1}} \left(\frac{\phi_{\gamma_t} \phi_{\gamma_{t+1}}}{n} W_{\sigma_{\gamma_t} \sigma_{\gamma_{t+1}}} \right) \prod_{s=t+1}^{\ell} A_{\gamma_s \gamma_{s+1}} \\ &= \Delta_{ef}^{(\ell)} + \sum_{t=0}^{\ell} \sum_{\gamma \in F_{ef}^{\ell+1}} \prod_{s=0}^{t-1} \underline{A}_{\gamma_s \gamma_{s+1}} \left(\frac{\phi_{\gamma_t} \phi_{\gamma_{t+1}}}{n} W_{\sigma_{\gamma_t} \sigma_{\gamma_{t+1}}} \right) \prod_{s=t+1}^{\ell} A_{\gamma_s \gamma_{s+1}}. \end{aligned} \tag{41}$$

Consider the two products in the summation over $F_{ef}^{\ell+1}$ on the right of (41): We can, for $1 \leq t \leq \ell - 1$, replace the summation over $F_{ef}^{\ell+1}$ by summing over all pairs $\gamma' = (\gamma_0, \ldots, \gamma_t) \in F_{eg}^{t}$ and $\gamma'' = (\gamma_{t+1}, \ldots, \gamma_{\ell+1}) \in F_{g'f}^{\ell-t}$ for some $g, g' \in \vec{E}(V)$ such that there exists a non-backtracking path with one intermediate edge, on the *complete* graph, between oriented edges g and g' (we denote this property by $g \xrightarrow{2} g'$). However caution is needed, as this summation also includes *tangled* paths, namely those in the sets $\{F_{t,ef}^{\ell+1}\}_{t=0}^{\ell}$. Where, for $1 \leq t \leq \ell - 1$,

$F^{\ell+1}_{t,ef}$ is defined as the collection of all *tangled* paths $\gamma = (\gamma_0, \dots, \gamma_{\ell+1}) = (\gamma', \gamma'') \in \Gamma^{\ell+1}_{ef}$ with γ' and γ'' as above. For $t = 0$, $F^{\ell+1}_{0,ef}$ consists of all non-backtracking *tangled* paths (γ', γ'') with $\gamma' = (e_1)$ and $\gamma'' \in F^{\ell}_{g'f}$ for any g' such that $g'_1 = e_2$. For $t = \ell$, $F^{\ell+1}_{\ell,ef}$ is the set of non-backtracking *tangled* paths (γ', γ'') such that $\gamma'' = (f_2)$ and $\gamma' \in F^{\ell}_{eg}$ for some $g \in \vec{E}(V)$ with $g_2 = f_1$. We rewrite (41) as

$$B^{(\ell)} = \Delta^{(\ell)} + \frac{1}{n} K B^{(\ell-1)} + \frac{1}{n} \sum_{t=1}^{\ell-1} \Delta^{(t-1)} K^{(2)} B^{(\ell-t-1)} + \frac{1}{n} \Delta^{(\ell-1)} \widehat{K} - \frac{1}{n} \sum_{t=0}^{\ell} R^{(\ell)}_t, \tag{42}$$

where for $e, f \in E_K$,

$$K_{ef} = 1_{e \to f} \phi_{e_1} \phi_{e_2} W_{\sigma(e_1)\sigma(e_2)}, \tag{43}$$

the *weighted* non-backtracking matrix on the *complete* graph (recall that $e \to f$ represents the non-backtracking property),

$$\widehat{K}_{ef} = 1_{e \to f} \phi_{f_1} \phi_{f_2} W_{\sigma(f_1)\sigma(f_2)}, \tag{44}$$

$$K^{(2)}_{ef} = 1_{e \overset{2}{\to} f} \phi_{e_2} \phi_{f_1} W_{\sigma(e_2)\sigma(f_1)}, \tag{45}$$

and where

$$(R^{(\ell)}_t)_{ef} = \sum_{\gamma \in F^{\ell+1}_{t,ef}} \prod_{s=0}^{t-1} \underline{A}_{\gamma_s \gamma_{s+1}} \phi_{\gamma_t} \phi_{\gamma_{t+1}} W_{\sigma(\gamma_t)\sigma(\gamma_{t+1})} \prod_{s=t+1}^{\ell} A_{\gamma_s \gamma_{s+1}}. \tag{46}$$

Indeed,

$$\begin{aligned} \left(\sum_{t=1}^{\ell-1} \Delta^{(t-1)} K^{(2)} B^{(\ell-t-1)} \right)_{ef} &= \sum_{t=1}^{\ell-1} \sum_{g,g'} \Delta^{(t-1)}_{eg} K^{(2)}_{gg'} B^{(\ell-t-1)}_{g'f} \\ &= \sum_{t=1}^{\ell-1} \sum_{g,g'} \sum_{\gamma' \in F^{t}_{eg}} \sum_{\gamma'' \in F^{\ell-t}_{g'f}} \prod_{s=0}^{t-1} \underline{A}_{\gamma'_s \gamma'_{s+1}} 1_{g \overset{2}{\to} g'} \phi_{\gamma'_t} \phi_{\gamma''_0} \\ &\quad \cdot W_{\sigma(\gamma'_t)\sigma(\gamma''_0)} \prod_{s=0}^{\ell-t-1} A_{\gamma''_s \gamma''_{s+1}}, \end{aligned} \tag{47}$$

$$\left(K B^{(\ell-1)} \right)_{ef} = \sum_{g} \sum_{\gamma'' \in F^{\ell}_{gf}} 1_{e \to g} \phi_{e_1} \phi_{e_2} W_{\sigma(e_1)\sigma(e_2)} A_{e_2, g_2} \prod_{s=1}^{\ell-2} A_{\gamma''_s \gamma''_{s+1}} A_{f_1 f_2}, \tag{48}$$

and,

$$\left(\Delta^{(\ell-1)} \widehat{K} \right)_{ef} = \sum_{g} \sum_{\gamma' \in F^{\ell}_{eg}} \underline{A}_{e_1 e_2} \prod_{s=1}^{\ell-2} \underline{A}_{\gamma'_s \gamma'_{s+1}} \underline{A}_{g_1 f_1} 1_{g \to f} \phi_{f_1} \phi_{f_2} W_{\sigma(f_1)\sigma(f_2)} \tag{49}$$

that is exactly the splitting described just below (41), where we also pointed out the need to compensate for *tangled* paths occuring in (47), which is precisely the role of $R^{(\ell)}_t$ in (42).

To bound (42), we introduce

$$\overline{W} = \frac{2}{\Phi^{(2)}} \left(\rho \chi_1 \check{\chi}_1^* + \mu_2 \chi_2 \check{\chi}_2^*\right) = \left(\phi_{e_2} \phi_{f_1} W_{\sigma(e_2)\sigma(f_1)}\right)_{ef}, \tag{50}$$

and,

$$L = K^{(2)} - \overline{W}. \tag{51}$$

Note the presence of weights in (50), hence our choice for the candidate eigenvectors.

Further, we set for $1 \leq t \leq \ell - 1$,

$$S_t^{(\ell)} = \Delta^{(t-1)} L B^{(\ell-t-1)}. \tag{52}$$

We then have:

▶ **Proposition 29** (Degree-Corrected Extension of Proposition 13 in [2]). *If G is tangle-free and $x \in \mathbb{C}^{\vec{E}(V)}$ with norm smaller than one, we have*

$$\|B^\ell x\| \leq \|\Delta^{(\ell)}\| + \frac{1}{n}\|KB^{(\ell-1)}\| + \frac{1}{n}\sum_{j=1,2}\frac{2\mu_j}{\Phi^{(2)}}\sum_{t=1}^{\ell-1}\|\Delta^{(t-1)}\chi_j\|\|\langle \check{\chi}_j, B^{\ell-t-1}x\rangle\|$$
$$+ \frac{1}{n}\sum_{t=1}^{\ell-1}\|S_t^{(\ell)}\| + \phi_{max}^2(a\vee b)\|\Delta^{(\ell-1)}\| + \frac{1}{n}\sum_{t=0}^{\ell}\|R_t^{(\ell)}\|.$$

Proof. Due to the tangle-freeness, $B^\ell = B^{(\ell)}$. Further $K^{(2)} = L + \overline{W}$ and $||K|| \leq \phi_{\max}^2(a \vee b)n$. ◀

In Appendix D in the detailed version of the underlying article (Arxiv:1609.02487) we prove the following bounds on the matrices in Proposition 29:

▶ **Proposition 30** (Degree-Corrected Extension of Proposition 14 in [2]). *Let $\ell = C\log_\rho n$ with $C < 1$. With high probability, the following norm bounds hold for all k, $0 \leq k \leq \ell$, and $i = 1, 2$:*

$$\|\Delta^{(k)}\| \leq (\log n)^{10}\rho^{k/2}, \tag{53}$$
$$\|\Delta^{(k)}\chi_i\| \leq (\log n)^5\rho^{k/2}\sqrt{n}, \tag{54}$$
$$\|R_k^{(\ell)}\| \leq (\log n)^{25}\rho^{\ell-k/2}, \tag{55}$$
$$\|KB^{(k)}\| \leq \sqrt{n}(\log n)^{10}\rho^k, \tag{56}$$

and the following bound holds for all k, $1 \leq k \leq \ell - 1$:

$$\|S_k^{(\ell)}\| \leq \sqrt{n}(\log n)^{20}\rho^{\ell-k/2}. \tag{57}$$

9.1 Proof of Proposition 5

From Propositions 29 and 30, the geometric growth in Corollary 21 together with the tangle-freeness due to Lemma 19, the proof of Proposition 5 follows:

Let $j \in \{1, 2\}$. If, for some vector x, $\langle \check{\varphi}_j, x\rangle = 0$, then $\langle B^\ell \chi_j, \check{x}\rangle = 0$. Therefore, using Corollary 21,

$$\begin{aligned}\sup_{\|x\|=1, \langle \check{\varphi}_j, x\rangle = 0} \langle \check{\chi}_j, B^{\ell-t-1}x\rangle &= \sup_{\|x\|=1, \langle B^\ell\chi_j, \check{x}\rangle = 0} \langle B^{\ell-t-1}\chi_j, \check{x}\rangle \\ &= \sup_{\|\check{x}\|=1, \langle B^\ell\chi_j, \check{x}\rangle = 0} \langle B^{\ell-t-1}\chi_j, \check{x}\rangle \\ &\leq \log^2(n) n^{1/2}\rho^{\frac{\ell-t-1}{2}}.\end{aligned} \tag{58}$$

With high probability, the graph is $\ell-$ tangle free (Lemma 19). Thus, invoking Propositions 29 and 30, with high probability,

$$\begin{aligned}\sup_{x\in H^{\perp},\|x\|=1}\|B^{\ell}x\| &\leq \log^{10}(n)\rho^{\frac{\ell}{2}}+n^{-1/2}\log^{10}(n)\rho^{\ell-1}\\ &\quad+c_1\log^{8}(n)\rho^{\frac{\ell}{2}}+n^{-1/2}\log^{21}(n)\rho^{\ell}\\ &\quad+c_2\log^{10}(n)\rho^{\frac{\ell}{2}}+n^{-1}\log^{26}(n)\rho^{\ell}\\ &\leq \log^{c}(n)\rho^{\frac{\ell}{2}},\end{aligned} \tag{59}$$

since $C<1$.

9.2 Comparison with the Stochastic Block Model in [2]

Putting $\phi_u=1$ for all u, we retrieve exactly **the same bounds as in the Stochastic Block Model**, that is equations $(30)-(34)$ in [2].

Below we use the trace method and therefore path counting combinatorial arguments to establish Proposition 30. In particular, we bound the expectation of expressions of the form

$$\mathbb{E}\left[\prod_{i=1}^{2m}\prod_{s=1}^{k}\underline{A}_{\gamma_{i,s-1}\gamma_{i,s}}\right], \tag{60}$$

for certain paths $\gamma=(\gamma_1,\ldots,\gamma_{2m})$ with $\gamma_i=(\gamma_{i,0},\cdots,\gamma_{i,k})\in V^{k+1}$, where $\underline{A}$ is defined in (38).

In bounding (60) the following term occurs:

$$\prod_{u\in V(\gamma)}\Phi^{(d_u)},$$

where $(d_u)_u$ are the degrees of the vertices in a specific tree (or forest) spanning the path γ. See, for instance, (D.4) and (D.17) in the detailed version of the underlying article (Arxiv:1609.02487). Here lies a major complication with respect to the Stochastic Block Model: those terms are not present in the latter model. In (D.8) and (D.19) in the detailed version of the underlying article, we find

$$\prod_{u=1}^{|V(\gamma)|}\Phi^{(d_u)}\leq C_2^{\sum_{u:d_u>2}(d_u-2)}\left(\Phi^{(2)}\right)^{|V(\gamma)|-n_{\mathcal{C}}},$$

where $C_2>1$ is some constant and where $n_{\mathcal{C}}\geq 1$ is the number of components on the path γ. To compare this term with powers of $\Phi^{(2)}$ (which are present in powers of $\rho=\frac{a+b}{2}\Phi^{(2)}$), we bound $\sum_{u:d_u>2}(d_u-2)$, see in particular Lemma (D.2) and (D.5) in the detailed version of the underlying article.

10 Detection: Proof of Theorem 2

The proofs of the statements in this section are deferred to Appendix E in the detailed version of the underlying article (Arxiv:1609.02487).

We need the following special case of a lemma in [2]:

▶ **Lemma 31** (Special case of Lemma 40 in [2]). *Assume that there exists a function $F : V \to \{0,1\}$ such that in probability, for any $i \in \{+,-\}$,*

$$\lim_{n\to\infty} \frac{1}{n} \sum_{v=1}^{n} 1_{\sigma(v)=i} F(v) = \frac{f(i)}{2},$$

where $f : \{+,-\} \to [0,1]$ is such that $f(+) > f(-)$. Then, assigning to each vertex a label $\widehat{\sigma}(v) = +$ if $F(v) = 1$ and $\widehat{\sigma}(v) = -$ if $F(v) = 0$, yields asymptotically positive overlap with the true spins.

Recall the eigenvector ξ_2 from Theorem 1. Below we use the function $F : v \mapsto 1_{\sum_{e:e_2=v} \xi_2(e) > \frac{\tau}{\sqrt{n}}}$ or $F : v \mapsto 1_{\sum_{e:e_2=v} \xi_2(e) \leq \frac{\tau}{\sqrt{n}}}$ for some fixed parameter τ. We verify also that ξ_2 is aligned with $P_{2,\ell}$. It is therefore useful to introduce the vector I_ℓ, defined element-wise by

$$I_\ell(v) = \sum_{e \in \vec{E} : e_2 = v} P_{2,\ell}(e), \tag{61}$$

for $v \in V$.

Further, put

$$\widehat{c} = \frac{a+b}{2} \frac{(\Phi^{(1)})^2 \Phi^{(3)}}{\Phi^{(2)}} \frac{\rho}{\mu_2^2 - \rho} \mu_2$$

The following lemma shows that I_ℓ is correlated with the spins:

▶ **Lemma 32** (Degree-Corrected Extension of Lemma 41 in [2]). *Let $\ell = C \log_\rho n$ with $C < C_{coupling}$ and $i \in \{+,-\}$. There exists a random variable Y_i such that $\mathbb{E}[Y_i] = 0$, $\mathbb{E}[|Y_i|] < \infty$ and for any continuity point t of the distribution of Y_i, in L^2,*

$$\frac{1}{n} \sum_{v=1}^{n} 1_{\sigma(v)=i} 1_{I_\ell(v)\mu_2^{-2\ell} - \widehat{c} g_2(i) \geq t} \to \frac{1}{2} \mathbb{P}(Y_i \geq t).$$

Recall from Theorem 1 that the eigenvector ξ_2 is asymptotically *aligned* with

$$\frac{B^\ell B^{*\ell} \check{\chi}_2}{\|B^\ell B^{*\ell} \check{\chi}_2\|}, \tag{62}$$

where $\ell \sim \log_\rho(n)$. Hence, for some unknown sign ω, the vector $\xi_2' = \omega \xi_2$ is asymptotically *close* to (62). From Lemma 28 we know that $B^\ell B^{*\ell} \check{\chi}_2$ and $P_{2,\ell}$ are asymptotically close. Consequently, properly renormalizing ξ_2' will make it asymptotically close to $P_{2,\ell}$, so that we can replace $P_{2,\ell}$ in (61) by ξ_2'. That is, we set for $v \in V$,

$$I(v) = \sum_{e:e_2=v} s\sqrt{n} \xi_2'(e),$$

with $s = \sqrt{c_2'''}$ the limit in Proposition 25. Then, I and $I_\ell/\mu_2^{2\ell}$ are close, which leads to the following lemma:

▶ **Lemma 33** (Degree-Corrected Extension of Lemma 42 in [2]). *Let $i \in \{+,-\}$ and $\widehat{Y}_i$ be as in Lemma 32. For any continuity point t of the distribution of $\widehat{Y}_i$, in L^2,*

$$\frac{1}{n} \sum_{v=1}^{n} 1_{\sigma(v)=i} 1_{I(v) - \widehat{c} g_2(i) \geq t} \to \frac{1}{2} \mathbb{P}\left(\widehat{Y}_i \geq t\right).$$

Put for $i \in \{+,-\}$, $X_i = \widehat{Y}_i + \widehat{c} g_2(i) = \widehat{Y}_i + \frac{1}{\sqrt{2}}\widehat{c} i$. Then, for all $t \in \mathbb{R}$ that are continuity points of the distribution of X_i, the following convergence holds in probability

$$\frac{1}{n}\sum_{v=1}^{n} 1_{\sigma(v)=i} 1_{I(v)>t} \to \frac{1}{2}\mathbb{P}\left(X_i > t\right).$$

Since $\mathbb{E}\left[X_+\right] > 0$, the argument below (90) in [2] establishes the existence of a continuity point $t_0 \in \mathbb{R}$ such that $\mathbb{P}\left(X_+ > t_0\right) > \mathbb{P}\left(X_- > t_0\right)$.

Further, we note that X_+ is in distribution equal to $-X_-$, a fact that we use below.

We are now in a position to apply Lemma 31 and thereby finishing the proof of Theorem 2:

If $\omega = 1$, then we define F, for $v \in V$, by

$$F(v) = 1_{\sum_{e:e_2=v} \xi_2(e) > \frac{t_0}{s\sqrt{n}}} = 1_{I(v)>t_0}.$$

Then,

$$\lim_{n\to\infty} \frac{1}{n}\sum_{v=1}^{n} 1_{\sigma(v)=+} F(v) = \frac{1}{2}\mathbb{P}\left(X_+ > t_0\right) =: \frac{f(+)}{2},$$

and,

$$\lim_{n\to\infty} \frac{1}{n}\sum_{v=1}^{n} 1_{\sigma(v)=-} F(v) = \frac{1}{2}\mathbb{P}\left(X_- > t_0\right) =: \frac{f(-)}{2},$$

so that $f(+) > f(-)$ and Lemma 31 applies.

If, however, $\omega = -1$, then we define F, for $v \in V$, by

$$F(v) = 1_{\sum_{e:e_2=v} \xi_2(e) \le \frac{t_0}{s\sqrt{n}}} = 1_{-I(v)\le t_0}.$$

Then, this time,

$$\lim_{n\to\infty} \frac{1}{n}\sum_{v=1}^{n} 1_{\sigma(v)=+} F(v) = \lim_{n\to\infty} \frac{1}{n}\sum_{v=1}^{n} 1_{\sigma(v)=+} 1_{I(v)>-t_0} = \frac{1}{2}\mathbb{P}\left(X_+ > -t_0\right) =: \frac{f(+)}{2},$$

since $-t_0$ is a continuity point of X_+, which follows from the fact that X_+ is in distribution equal to $-X_-$ and t_0 is a continuity point of X_-.

Similarly,

$$\lim_{n\to\infty} \frac{1}{n}\sum_{v=1}^{n} 1_{\sigma(v)=-} F(v) = \frac{1}{2}\mathbb{P}\left(X_- > -t_0\right) =: \frac{f(-)}{2}.$$

Now,

$$f(+) = \mathbb{P}\left(X_+ > -t_0\right) = 1 - \mathbb{P}\left(X_- > t_0\right) > 1 - \mathbb{P}\left(X_+ > t_0\right) = \mathbb{P}\left(X_- > -t_0\right) = f(-),$$

exactly the setting of Lemma 31.

References

1 B. Bollobás, S. Janson, and O. Riordan. The phase transition in inhomogeneous random graphs. *Random Struct. Algorithms*, 31(1):3–122, August 2007.

2 C. Bordenave, M. Lelarge, and L. Massoulié. Non-backtracking spectrum of random graphs: community detection and non-regular ramanujan graphs. *arXiv preprint 1501.06087*, 2015.

3 A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Phys. Rev. E*, 84:066106, Dec 2011.

4 W. Evans, C. Kenyon, Y. Peres, and L. Schulman. Broadcasting on trees and the ising model. *Ann. Appl. Probab.*, 10(2):410–433, 05 2000.

5 Z. Füredi and J. Komlós. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3):233–241, 1981.

6 L. Gulikers, M. Lelarge, and L. Massoulié. An impossibility result for reconstruction in a degree-corrected planted-partition model. *arXiv preprint 1511.00546*, 2015.

7 L. Gulikers, M. Lelarge, and L. Massoulié. A spectral method for community detection in moderately-sparse degree-corrected stochastic block models. *arXiv preprint 1506.08621*, 2015.

8 P. Holland, K. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, June 1983.

9 M. Horton, H. Stark, and T. Terras. What are zeta functions of graphs and what are they good for? *Contemporary Mathematics, Quantum Graphs and Their Applications;*, 415:173–190, 2006.

10 K. i. Hashimoto. Zeta functions of finite graphs and representations of p-adic groups. In *Automorphic Forms and Geometry of Arithmetic Varieties*, volume 15 of *Advanced Studies in Pure Mathematics*, pages 211 – 280. Academic Press, 1989.

11 B. Karrer and M. Newman. Stochastic blockmodels and community structure in networks. *Phys. Rev. E*, 83:016107, Jan 2011.

12 H. Kesten and B. P. Stigum. Additional limit theorems for indecomposable multidimensional galton-watson processes. *Ann. Math. Statist.*, 37(6):1463–1481, 12 1966.

13 H. Kesten and B. P. Stigum. A limit theorem for multidimensional galton-watson processes. *Ann. Math. Statist.*, 37(5):1211–1223, 10 1966.

14 F. Krzakala, C. Moore, E. Mossel, J. Neeman, A. Sly, L. Zdeborová, and P. Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, 110(52):20935–20940, 2013.

15 A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

16 L. Massoulié. Community detection thresholds and the weak ramanujan property. *ACM Symposium on the Theory of Computing (STOC)*, 2014.

17 E. Mossel, J. Neeman, and A. Sly. A proof of the block model threshold conjecture. *arXiv preprint 1311.4115*, 2015.

18 E. Mossel, J. Neeman, and A. Sly. Reconstruction and estimation in the planted partition model. *Probability Theory and Related Fields*, 162(3):431–461, 2015.

19 R. Murty. Ramanujan graphs. *J. Ramanujan Math. Soc.*, 18(1):1–20, 2003.

Conditional Sparse Linear Regression

Brendan Juba*

Washington University, St. Louis, USA
bjuba@wustl.edu

Abstract

Machine learning and statistics typically focus on building models that capture the vast majority of the data, possibly ignoring a small subset of data as "noise" or "outliers." By contrast, here we consider the problem of *jointly* identifying a significant (but perhaps small) segment of a population in which there is a highly sparse linear regression fit, together with the coefficients for the linear fit. We contend that such tasks are of interest both because the models themselves may be able to achieve better predictions in such special cases, but also because they may aid our understanding of the data. We give algorithms for such problems under the sup norm, when this unknown segment of the population is described by a k-DNF condition and the regression fit is s-sparse for constant k and s. For the variants of this problem when the regression fit is *not* so sparse or using expected error, we also give a preliminary algorithm and highlight the question as a challenge for future work.

1998 ACM Subject Classification G.3 Probability and Statistics, F.2.0 Analysis of Algorithms and Problem Complexity: General

Keywords and phrases Linear regression, conditional regression, conditional distribution search

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.45

1 Introduction

Linear regression, the fitting of linear relationships among variables in a data set, is a standard tool in data analysis. In particular, for the sake of interpretability and utility in further analysis, we desire to find *highly sparse* linear relationships, i.e., involving only a few variables. Of course, such simple linear relationships often will not hold across an entire population. But, more frequently there will exist conditions – perhaps a range of parameters or a segment of a larger population – under which such sparse models fit the data quite well. For example, Rosenfeld et al. [22] used data mining heuristics to identify small segments of a population in which a few additional risk factors were highly predictive of certain kinds of cancer, whereas these same risk factors were not significant in the overall population. Simple rules for special cases may also hint at the more complex general rules. More generally, we need to develop new techniques to reason about populations in which most members are atypical in some way, which are colloquially (and somewhat abusively) referred to as *long-tailed* distributions. We are seeking computationally efficient, principled alternatives to ad-hoc approaches such as trying a variety of methods for clustering the data and hoping that the identified clusters can be modeled well.

* Supported by an AFOSR Young Investigator Award.

© Brendan Juba;
licensed under Creative Commons License CC-BY

1.1 Our results

In this work we consider the design and analysis of efficient algorithms for the *joint* task of identifying significant segments of a population in which a sparse model provides a good fit. We are able to identify such segments when they are described by a k-DNF and there is a s-sparse regression fit for constant k and s. More specifically, we give algorithms when there is a linear relationship with respect to which the error is bounded by ϵ with probability 1 (i.e., ϵ sup norm). In this case, we find a condition in which the error is bounded by ϵ for a $1-\gamma$ fraction of the population (with probability $1-\delta$ over the sample of data).

▶ **Theorem 1** (Conditional sparse linear regression). *Suppose that D is a joint probability distribution over $\vec{x} \in \{0,1\}^n$, $\vec{y} \in \mathbb{R}^d$, and $z \in \mathbb{R}$ such that there is a k-DNF c for which for some s-sparse $\vec{a} \in \mathbb{R}^d$*

$$\Pr_{(x,y,z)\in D}\left[|\langle \vec{a}, \vec{y}\rangle - z| \le \epsilon \middle| c(\vec{x}) = 1\right] = 1 \qquad \text{and} \qquad \Pr_{(x,y,z)\in D}[c(\vec{x}) = 1] \ge \mu.$$

Then given ϵ, μ, and δ in $(0,1)$, $\gamma \in (0, 1/2]$, and access to examples from D, for any constants s and k, there is an algorithm that runs in polynomial time in n, d, $1/\mu$, $1/\gamma$, and $\log 1/\delta$, and finds an s-sparse $\vec{a}'$ and k-DNF c' such that with probability $1-\delta$,

$$\Pr_{(x,y,z)\in D}\left[|\langle \vec{a}', \vec{y}\rangle - z| \le \epsilon \middle| c'(\vec{x}) = 1\right] \ge 1-\gamma \qquad \text{and} \qquad \Pr_{(x,y,z)\in D}[c'(\vec{x}) = 1] \ge (1-\gamma)\mu.$$

Our algorithms make crucial use of the sought solution's sparsity. The key observation is that since the linear rule has constant sparsity, with respect to the relevant dimensions there are a constant number of "extremal examples" such that we can obtain low error on the unknown event by fitting these extremal examples. We can then use the linear rule we obtain from fitting such a set of examples to label the data according to whether or not that point has low error under the linear rule. Finally, this enables us to find an event on which the linear rule has low error. Thus, it suffices to simply perform a search over candidates for the extremal examples and return one for which the corresponding event captures enough of the data.

We also note a trivial (weak) approximation algorithm for an expected-error variant of the problem that does not rely on sparsity: when there is a k-DNF c and a linear rule a giving conditional expected error ϵ (and c is true with probability μ), we find a condition c' and a linear rule a' with conditional expected error $O(n^k\epsilon)$ and probability $\Omega(\mu/n^k)$. We pose the design of better algorithms for the dense regression and expected-error tasks as challenges for future work.

1.2 Related work

We are building on recent work by Juba [15] on identifying potentially rare events of interest in a distribution, which captures a family of data mining tasks similar to, e.g., association rule discovery [1] or "bump hunting" [10]. This work is closely related to theoretical work on *positive-reliable learning* [16, 17], which is in turn very closely related to the "heuristic learning" model introduced by Pitt and Valiant [21] and studied in depth by Bshouty and Burroughs [5]: these are models of classification in which one type of error is minimized subject to a hard bound on the other type of error. The key difference between heuristic or positive-reliable learning on the one hand, and the work by Juba or the work in data mining on the other, is that the latter works focus on bounding the error *conditioned on the identified event* (i.e., the *precision* rather than the raw *false-positive rate*). In the present

work, we develop this perspective further, and seek to perform *supervised learning* in such a conditional distribution. In this context, these earlier works can be viewed as identifying a conditional distribution in which the class consisting solely of the constant 1 function fits the selected points with low error. We are generalizing this to the problem of fitting a (sparse) linear rule in the identified conditional distribution.

Our work also has some relationship to the enormous body of work on *robust statistics* [13, 23], in which *outliers* are identified and ignored or otherwise mitigated. The difference in what we consider here is two-fold. First, we are specifically interested in the case where *we may decline to fit the vast majority of the data,* thus treating most of the data as "outliers" in the model of robust statistics. Second, we are also interested in *finding a (simple) rule that identifies which subset of the data we are fitting* (and which subset we are ignoring). By contrast, in robust statistics, an arbitrary subset of the data may be considered "corrupted" and ignored. Note that without this extra structure, Hardt and Moitra [12] found that the problem of finding subspaces described by a single linear constraint is intractable (precisely, SSE-hard) when the subspace does not contain nearly all $(1 - 1/d$ fraction) of the data.

Similarly, our problem differs from linear mixed models [18, 14] in that linear mixed models seek several linear rules to try to explain (almost) all of the data. Again, in such models, the only description of the "clusters" are the linear models themselves (so points are taken to lie in the cluster of the linear fit with which they have the smallest residual).

Our problem is also very closely related to the problem solved by RANSAC [9] and its variants, that use sampling to find nontrivial linear relationships in data even when these are only of moderate density. The difference is principally that RANSAC is designed to find linear relationships in very low dimension (e.g., in $\mathbb{R}^2$), and does not scale to high dimensions since we need d points to determine a linear fit in $\mathbb{R}^d$, i.e., we need to hit the subspace d times when sampling. In the present work, by contrast, although the linear fit we are seeking is of constant sparsity, we wish to find linear relationships in asymptotically growing dimension d. Also, RANSAC-like algorithms, as in robust statistics (or linear mixed models), do not aim to provide a description of the data for which they find a linear relationship.

Finally, we note that our work has a connection to the list-learning model introduced independently (subsequent to the original posting of this work on arXiv) by Charikar et al. [6]. The connection is more technical, and we postpone discussing it to Section 5.

2 Problem definition and background

In this work, we primarily focus on the following task:

▶ **Definition 2** (Conditional linear regression)**.** The *conditional (realizable) linear regression* task is the following. We are given access to examples from an arbitrary distribution D over $\{0,1\}^n \times \mathbb{R}^d \times \mathbb{R}$ for which there exists a k-DNF c^* and $\vec{a}^* \in \mathbb{R}^d$ such that

1. $\Pr_{(x,y,z)\in D}[|\langle \vec{a}^*, \vec{y}\rangle - z| \leq \epsilon | c^*(\vec{x}) = 1] = 1$ and
2. $\Pr_{(x,y,z)\in D}[c^*(\vec{x}) = 1] \geq \mu$,

for some $\epsilon, \mu \in (0,1]$. Then with probability $1 - \delta$, given ϵ, μ, δ, and γ as input, we are to find some $\vec{a}' \in \mathbb{R}^d$ and k-DNF c' such that

1. $\Pr_{(x,y,z)\in D}[|\langle \vec{a}', \vec{y}\rangle - z| \leq \epsilon | c'(\vec{x}) = 1] \geq 1 - \gamma$ and
2. $\Pr_{(x,y,z)\in D}[c'(\vec{x}) = 1] \geq \Omega\left(\left((1-\gamma)\frac{\mu}{nd}\right)^k\right)$ for some k

in time polynomial in $n, d, 1/\mu, 1/\epsilon, 1/\gamma$, and $1/\delta$. If $\vec{a}^*$ is assumed to have at most s nonzero entries and $\vec{a}'$ is likewise required to have at most s nonzero entries, then this is the *conditional sparse linear regression* task with *sparsity* s.

We will also briefly consider the following variant that in some contexts may be more natural.

▶ **Definition 3** (Conditional ℓ_2-linear regression)**.** The *conditional ℓ_2-linear regression* task is the following. We are given access to examples from an arbitrary distribution D over $\{0,1\}^n \times \{\vec{y} \in \mathbb{R}^d : \|\vec{y}\|_2 \leq B\} \times [-B, B]$ for which there exists a k-DNF c^* and $\vec{a}^* \in \mathbb{R}^d$ with $\|\vec{a}^*\|_2 \leq B$ such that

1. $\mathbb{E}_{(x,y,z)\in D}\left[(\langle \vec{a}^*, \vec{y}\rangle - z)^2 | c^*(\vec{x}) = 1\right] \leq \epsilon$ and
2. $\Pr_{(x,y,z)\in D}[c^*(\vec{x}) = 1] \geq \mu$,

for some $B \in \mathbb{R}^+$, $\epsilon, \mu \in (0, 1]$. Then with probability $1 - \delta$, given B, ϵ, μ, δ, and γ as input, we are to find some $\vec{a}' \in \mathbb{R}^d$ and k-DNF c' such that

1. $\mathbb{E}_{(x,y,z)\in D}\left[(\langle \vec{a}', \vec{y}\rangle - z)^2 | c'(\vec{x}) = 1\right] \leq \mathrm{poly}(B, d, n)\epsilon$ and
2. $\Pr_{(x,y,z)\in D}[c'(\vec{x}) = 1] \geq \Omega\left(\left((1-\gamma)\frac{\mu}{Bdn}\right)^k\right)$ for some k

in time polynomial in $n, d, B, 1/\mu, 1/\epsilon, 1/\gamma$, and $1/\delta$.

One could further consider, for example, regression under the other ℓ_p norms, but we will not pursue this here.

In both variants of the problem, we have sought to only recover a condition c' that only comprises a polynomial fraction of the probability of the optimal condition μ. A controllable $1 - \gamma$ factor loss is generally necessary when we are choosing among various candidate "clusters" based on sampling. Although in the earlier work on conditional distribution search [15] (see Definition 4, next), it was possible to find an event c' that actually captured the same probability mass as the target c (since there we are uniquely seeking a "cluster" that selects positive points), even in that setting, the reductions between related models generally incurred a controllable $1-\gamma$ loss. Nevertheless, the value of such a formal definition is usually in enabling us to formulate negative results, and in that case we seek the most liberal definition possible. Hence, here, we allow c' to only contain a polynomial fraction of μ depending on the main parameters, B, d, and n, that we might expect to encounter in an approximation guarantee, such as the one we show for Algorithm 2.

The restriction of c to be a k-DNF is not arbitrary. Although we could consider other classes of representations for c, it seems that essentially any of the other standard hypothesis classes that we might naturally consider here will lead to an intractable problem, even under the relatively liberal version of the problems defined above. This will follow since we can reduce the simpler problem of finding such conditions to our problem:

▶ **Definition 4** (Conditional distribution search)**.** For a *representation class* $\mathcal{C}$ of $c : \{0,1\}^n \to \{0,1\}$, the *conditional distribution search problem* is as follows. Given access to i.i.d. examples $(\vec{x}^{(1)}, b^{(1)}), \ldots, (\vec{x}^{(m)}, b^{(m)})$ from an arbitrary distribution D over $\{0,1\}^n \times \{0,1\}$ for which there exists $c^* \in \mathcal{C}$ such that $\Pr_{(x,b)\in D}[b = 1 | c^*(\vec{x}) = 1] = 1$ and $\Pr_{(x,b)\in D}[c^*(\vec{x}) = 1] \geq \mu$, with probability $1 - \delta$, find some circuit c' such that

1. $\Pr_{(x,b)\in D}[b = 1 | c'(\vec{x}) = 1] \geq 1 - \gamma$ and
2. $\Pr_{(x,b)\in D}[c'(\vec{x}) = 1] \geq \Omega(((1-\gamma)\mu/n)^k)$ for some k

in time polynomial in $n, 1/\mu, 1/\gamma$, and $1/\delta$.

▶ **Theorem 5** (Conditional distribution search reduces to conditional linear regression)**.** *Suppose there is an algorithm that given access to examples from an arbitrary distribution D' over $\{0,1\}^n \times \{0,1\} \times \{0,1\}$ for which there exists $c^* \in \mathcal{C}$ and $a^* \in \mathbb{R}$ such that*

$$\Pr_{(x,y,z)\in D'}\left[|a^* y - z| \leq \epsilon \middle| c^*(\vec{x}) = 1\right] = 1 \text{ and } \Pr_{(x,y,z)\in D'}[c^*(\vec{x}) = 1] \geq \mu,$$

with probability $1-\delta$*, finds some* $a' \in \mathbb{R}$ *and circuit* c' *such that*

$$\Pr_{(x,y,z)\in D'}\left[|a'y - z| \leq \epsilon \middle| c'(\vec{x}) = 1\right] \geq 1-\gamma \qquad \textit{and}$$

$$\Pr_{(x,y,z)\in D'}[c'(\vec{x}) = 1] \geq \Omega\left(\left(\frac{(1-\gamma)\mu}{n}\right)^k\right) \qquad \textit{for some } k$$

in time polynomial in $n, 1/\mu, 1/\gamma, 1/\epsilon$ *and* $1/\delta$*. Then there is a randomized polynomial-time algorithm for conditional distribution search for* $\mathcal{C}$*.*

Proof. Let D be a distribution satisfying the hypotheses of the conditional distribution search task for $\mathcal{C}$, that is, for some $c^* \in \mathcal{C}$,

1. $\Pr_{(x,b)\in D}[b = 1 | c^*(\vec{x}) = 1] = 1$ and
2. $\Pr_{(x,b)\in D}[c^*(\vec{x}) = 1] \geq \mu$.

Let D' be the distribution over $\{0,1\}^n \times \{0,1\} \times \{0,1\}$ sampled as follows: given an example $(\vec{x}, b)$ from D, if $b = 1$ we produce $(\vec{x}, 1, 0)$ and otherwise we produce $(\vec{x}, 1, b')$ for b' uniformly distributed over $\{0,1\}$. Notice that for c^* and $a^* = 0$, then whenever $c^*(\vec{x}) = 1$, $|a^*y - z| = 0 \leq 1/3$ over the entire support of the distribution; and, by assumption, $\Pr_{(x,y,z)\in D'}[c^*(\vec{x}) = 1] = \Pr_{(x,b)\in D}[c^*(\vec{x}) = 1] \geq \mu$. So, the pair $a^* = 0$ and c^* certainly satisfy the conditions for our task for $\epsilon = 1/3$. Therefore, by hypothesis, an algorithm for our task given access to D' with $\epsilon = 1/3$ and $\gamma' = \gamma/2$ must return a' and a circuit c' such that

1. $\Pr_{(x,y,z)\in D'}\left[|a'y - z| \leq 1/3 | c'(\vec{x}) = 1\right] \geq 1 - \gamma'$ and
2. $\Pr_{(x,y,z)\in D'}[c'(\vec{x}) = 1] \geq \Omega(((1-\gamma')\mu/n)^k)$ for some k.

But now, since the distribution we used is uniform over examples with $z = 0$ and $z = 1$ whenever $b = 0$ (and $y \equiv 1$), it must be that whatever a' is returned, $|a' - z| > 1/3$ with probability $1/2$ conditioned on $b = 0$ in the underlying draw from D. We must therefore actually have that

$$\frac{1}{2}\Pr_{(x,b)\in D}[b = 0 | c'(\vec{x}) = 1] \leq \Pr_{(x,y,z)\in D'}\left[|a'y - z| > 1/3 \middle| c'(\vec{x}) = 1\right] \leq \frac{\gamma}{2}$$

so indeed, also $\Pr_{(x,b)\in D}[b = 1 | c'(\vec{x}) = 1] \geq 1 - \gamma$. Thus c' is as needed for a solution to the conditional distribution search problem. Since it is trivial to implement the sampling oracle for D' given a sampling oracle for D, we obtain the desired algorithm. ◀

In turn now, algorithms for finding such conditions would yield algorithms for PAC-learning DNF [15], which is currently suspected to be intractable (c.f. in particular work by Daniely and Shalev-Shwartz [8] for some strong consequences of learning DNF).

▶ **Theorem 6** (Theorem 5 of [15])**.** *If there exists an algorithm for the conditional distribution search problem for conjunctions, then DNF is PAC-learnable in polynomial time.*

Informally, therefore, an algorithm for conditional realizable linear regression for conjunctions, or any class that can *express* conjunctions (instead of k-DNF), even under the relatively lax version of the problem formulated here, would yield a randomized polynomial time algorithm for PAC-learning DNF. This seems to rule out, in particular, the possibility of developing algorithms to perform regression under conditions described by halfspaces, decision trees, and so on.

For conditional ℓ_2-linear regression, a stronger conclusion holds: such algorithms would solve the *agnostic* variant of the conditional distribution search task, with a similar error bound:

▶ **Theorem 7** (Agnostic condition search reduces to conditional ℓ_2-linear regression)**.** *Suppose there is an algorithm that given access to examples from an arbitrary distribution D' over $\{0,1\}^n \times \{0,1\} \times \{0,1\}$ for which there exists $c^* \in \mathcal{C}$ and $a^* \in [0,1]$ such that $\mathbb{E}_{(x,y,z)\in D'}\left[(a^*y - z)^2 | c^*(\vec{x}) = 1\right] \leq \epsilon$ and $\Pr_{(x,y,z)\in D'}[c^*(\vec{x}) = 1] \geq \mu$, with probability $1-\delta$, finds some a' and circuit c' such that*
1. $\mathbb{E}_{(x,y,z)\in D'}\left[(a'y - z)^2 | c'(\vec{x}) = 1\right] \leq p(n)\epsilon$ *for some polynomial p and*
2. $\Pr_{(x,y,z)\in D'}[c'(\vec{x}) = 1] \geq \Omega(((1-\gamma)\mu/n)^k)$ *for some k*
in time polynomial in $n, 1/\mu, 1/\gamma, 1/\epsilon$ and $1/\delta$. Then there is a randomized polynomial-time algorithm for agnostic conditional distribution search for $\mathcal{C}$: that is, if there exists $c \in \mathcal{C}$ achieving
1. $\Pr_{(x,b)\in D}[b = 1 | c(\vec{x}) = 1] \geq 1 - \epsilon$ *and*
2. $\Pr_{(x,b)\in D}[c(\vec{x}) = 1] \geq \mu$
then the algorithm finds a circuit c'' achieving
1. $\Pr_{(x,b)\in D}[b = 1 | c''(\vec{x}) = 1] \geq 1 - 2p(n)\epsilon$ *and*
2. $\Pr_{(x,b)\in D}[c''(\vec{x}) = 1] \geq \Omega(((1-\gamma)\mu/n)^k)$ *for some k*
in time polynomial in $n, 1/\mu, 1/\gamma, 1/\epsilon$ and $1/\delta$.

Proof. For a given distribution D over (x, b) satisfying the promise for conditional distribution search, we use the same construction of D' and reduction as in the proof of Theorem 5. Here, we note that for $a^* = 0$, given that $\Pr_{(x,b)\in D}[b = 1 | c(\vec{x}) = 1] \geq 1 - \epsilon$ for the c assumed to exist for conditional distribution search

$$\mathop{\mathbb{E}}_{(x,y,z)\in D'}\left[(0 \cdot 1 - z)^2 \middle| c(\vec{x}) = 1\right] \leq \frac{1}{2}\epsilon.$$

Therefore, an algorithm for conditional ℓ_2-linear regression must find some a' and circuit c' such that $\Pr_{(x,y,z)\in D'}[c'(\vec{x}) = 1] \geq \Omega(((1-\gamma)\mu/n)^k)$ for some k and

$$\mathop{\mathbb{E}}_{(x,y,z)\in D'}\left[((a' - z)^2 \middle| c'(\vec{x}) = 1\right] \leq \frac{1}{2}p(n)\epsilon.$$

Now, again, since D' gives $z = 0$ and $z = 1$ equal probability whenever $b = 0$, we note that for such examples the expected value of $(a' - z)^2$ is minimized by $a' = 1/2$, where it achieves expected value $1/4$. Thus as $(a' - z)^2$ is surely nonnegative,

$$\frac{1}{4}\Pr_{(x,b)\in D}[b = 0 | c'(\vec{x}) = 1] \leq \mathop{\mathbb{E}}_{(x,y,z)\in D'}\left[(a' - z)^2 \middle| c'(\vec{x}) = 1\right] \leq \frac{1}{2}p(n)\epsilon$$

so c' indeed also achieves $\Pr_{(x,b)\in D}[b = 1 | c'(\vec{x}) = 1] \geq 1 - 2p(n)\epsilon$. ◀

The restriction to constant sparsity is also key, as our problem contains as a special case (when $\mu = 1$, that is, when the trivial condition that takes the entire population can be used) the standard sparse linear regression problem. Sparse linear regression for *constant* sparsity is easy, but when the sparsity is allowed to be large, the problem quickly becomes intractable: In general, finding sparse solutions to linear equations is known to be NP-hard [20], and Zhang, Wainwright, and Jordan [30] extend this to bounds on the quality of sparse linear regression that is achievable by polynomial-time algorithms, given that NP does not have polynomial-size circuits.

3 Algorithms for conditional sparse linear regression

We now turn to stating and proving our main theorem. In what follows, we use the following (standard) notation: $\Pi_{d_1,\ldots,d_s}$ denotes the projection (of $\mathbb{R}^d$) to the s coordinates $d_1, d_2, \ldots, d_s$

Algorithm 1: Find-and-eliminate.

input : Examples $(\vec{x}^{(1)}, \vec{y}^{(1)}, z^{(1)}), \ldots, (\vec{x}^{(m)}, \vec{y}^{(m)}, z^{(m)})$, target fit ϵ and fraction $(1-\gamma/2)\mu$.

output : A k-DNF over $x_1, \ldots, x_n$ and linear predictor over $y_1, \ldots, y_d$, or INFEASIBLE if none exist.

begin

  forall $(d_1, \ldots, d_s) \in \binom{[d]}{s}$, $(\sigma_1, \ldots, \sigma_{s+1}) \in \{\pm 1\}^{s+1}$ *and* $(j_1, \ldots, j_{s+1}) \in \binom{[m]}{s+1}$ **do**

    Initialize c to be the (trivial) k-DNF over all $\binom{2n}{k}$ terms of size k.

    Let $(\vec{a}, \epsilon')$ be a solution to the following linear system:

$$\langle \vec{a}, \Pi_{d_1,\ldots,d_s} \vec{y}^{(j_\ell)} \rangle - z^{(j_\ell)} = \sigma_\ell \epsilon' \text{ for } \ell = 1, \ldots, s+1$$

    if $\epsilon' > \epsilon$ **then** continue to the next iteration.

    for $j = 1, \ldots, m$ **do** **if** $|\langle \vec{a}, \Pi_{d_1,\ldots,d_s} \vec{y}^{(j)} \rangle - z^{(j)}| > \epsilon$ **then**

      forall $T \in c$ **do** **if** $T(\vec{x}^{(j)}) = 1$ **then** Remove T from c.

    end

    if $\#\{j : c(\vec{x}^{(j)}) = 1\} > (1-\gamma/2)\mu m$ **then** **return** $\vec{a}$ *and* c.

  end

  return *INFEASIBLE.*

end

from $[d]$ (which denotes the integers $1, \ldots d$). For a set S, we let $\binom{S}{k}$ denote the subsets of S of size exactly k.

At a high level, the algorithm (Algorithm 1, below) generates a *list* of possible coefficient vectors for the regression fit. For each such candidate, it generates labels for the points indicating whether or not the candidate linear fit achieves small error under that fit or not. It then solves the conditional distribution search problem given by these labels (by using the Elimination algorithm [15]), and estimates the fraction of the data captured this way. It returns the first linear fit that captures a sufficiently large fraction (or "INFEASIBLE" if none do).

▶ **Theorem 8** (Realizable sparse regression – full statement of Theorem 1). *Suppose that D is a joint probability distribution over $\vec{x} \in \{0,1\}^n$, $\vec{y} \in \mathbb{R}^d$, and $z \in \mathbb{R}$ such that there is a k-DNF c for which for some s-sparse $\vec{a} \in \mathbb{R}^d$*

$$\Pr_{(x,y,z)\in D}\left[|\langle \vec{a}, \vec{y}\rangle - z| \le \epsilon \middle| c(\vec{x}) = 1\right] = 1 \qquad \text{and} \qquad \Pr_{(x,y,z)\in D}[c(\vec{x}) = 1] \ge \mu.$$

Then given ϵ, μ, and δ in $(0,1)$ and $\gamma \in (0, 1/2]$ and

$$m = O\left(\frac{1}{\mu\gamma}\left(s\log s + s\log d + n^k + \log\frac{1}{\delta}\right)\right)$$

examples from D, for any constants s and k, Algorithm 1 runs in polynomial time in n, d, and m $(= \mathrm{poly}(n, d, 1/\mu, 1/\gamma, \log 1/\delta))$ and finds an s-sparse $\vec{a}'$ and k-DNF c' such that with probability $1-\delta$,

$$\Pr_{(x,y,z)\in D}\left[|\langle \vec{a}', \vec{y}\rangle - z| \le \epsilon \middle| c'(\vec{x}) = 1\right] \ge 1-\gamma \qquad \text{and} \qquad \Pr_{(x,y,z)\in D}[c'(\vec{x}) = 1] \ge (1-\gamma)\mu.$$

Proof. It is clear that the algorithm runs for $O(d^s m^{s+1})$ iterations, where each iteration (for constant s) runs in time polynomial in the bit length of our examples and $O(mn^k)$. Thus, for constant s and k, the algorithm runs in polynomial time overall, and it only remains to argue correctness.

We will first argue that the algorithm succeeds at returning some solution with probability $1 - \delta/3$ over the examples. We will then argue that any solution returned by the algorithm is satisfactory with probability $1 - 2\delta/3$ over the examples, thus leading to a correct solution with probability $1 - \delta$ overall.

Completeness part 1: Generating the linear rule

We first note that for $m \geq \frac{6}{\mu\gamma} \ln \frac{3}{\delta}$ examples, a Chernoff bound guarantees that with probability $1-\delta/3$, there are at least $(1-\gamma/2)\mu m$ examples satisfying the unknown condition c in the sample. Let S be the set of examples satisfying c. Given the set of s dimensions that are used in the sparse linear rule, we set up a linear program in $s+1$ dimensions to minimize ϵ' subject to the constraints

$$-\epsilon' \leq \langle \vec{a}, \vec{y}^{(j)} \rangle - z^{(j)} \leq \epsilon' \text{ for } j \in S.$$

It is well known (see, for example, Schrijver [24, Chapter 8]) that the optimum value for any feasible linear program may be obtained at a *basic feasible solution,* i.e., a vertex of the polytope, given by satisfying $s+1$ of the constraints with equality. Since each constraint corresponds to an example and sign (for the lower or upper inequality), this means that we can obtain $\vec{a}$ by solving for $\vec{a}$ and ϵ' in the following linear system

$$\langle \vec{a}, \vec{y}^{(j_\ell)} \rangle - z^{(j_\ell)} = \sigma_\ell \epsilon' \text{ for } \ell = 1, \ldots, s+1$$

for some set of $s+1$ examples, $j_1, \ldots, j_{s+1}$ and $s+1$ signs $\sigma_1, \ldots, \sigma_{s+1}$ corresponding to the tight constraints. Thus, when the algorithm uses the appropriate set of s dimensions, the appropriate $s+1$ examples, and the appropriate $s+1$ signs, we will recover an $\vec{a}^*$ and ϵ^* such that for all $j \in S$, $|\langle \vec{a}^*, \vec{y}^{(j)} \rangle - z^{(j)}| \leq \epsilon^* \leq \epsilon$.

Completeness part 2: Recovering a suitable condition given a rule

Now, given $\vec{a}^*$ such that for all $j \in S$, $|\langle \vec{a}^*, \vec{y}^{(j)} \rangle - z^{(j)}| \leq \epsilon$, we observe that the algorithm identifies a k-DNF h^* such that $h^*(\vec{x}^{(j)}) = 1$ for all $j \in S$. Indeed, the algorithm only eliminates a k-term T for examples j such that $|\langle \vec{a}^*, \vec{y}^{(j)} \rangle - z^{(j)}| > \epsilon$. Thus, it never eliminates any term appearing in c, and so in particular, $\Pr_{(x,y,z)\in D}[h^*(\vec{x}) = 1] \geq \Pr_{(x,y,z)\in D}[c(\vec{x}) = 1] \geq \mu$. Moreover, since (as noted above, with probability $1 - \delta/3$) there are at least $(1-\gamma/2)\mu m$ examples satisfying c in the sample, there are at least $(1-\gamma/2)\mu m$ examples satisfying h^*. Thus, with probability $1 - \delta/2$, when the algorithm considers the relevant s dimensions in the support of $\vec{a}$ and considers an appropriate choice of $s+1$ examples to obtain a suitable $\vec{a}^*$, it will furthermore obtain an h^* that will lead the algorithm to terminate and return $\vec{a}^*$ and h^*.

Soundness: Generalization bounds

Next, we argue that any $\vec{a}'$ and h' returned by the algorithm will suffice with probability $1 - 2\delta/3$ over the examples.

We will use the facts that

1. a union of k hypothesis classes of VC-dimension d has VC-dimension at most $O(d \log d + \log k)$ (for example, see [25, Exercise 6.11]),
2. linear threshold functions in $\mathbb{R}^s$ have VC-dimension $s+1$ (e.g., [25, Section 9.1.3]), and
3. the composition of classes of VC-dimension d_1 and d_2 has VC-dimension at most $d_1 + d_2$ (follows from [25, Exercise 20.4]).

We now consider the class of disjunctions of a k-CNF over $\{0,1\}^n$ and (intersections of) linear threshold functions $[|\langle(\vec{a},-1),(\vec{y},z)\rangle| \leq \varepsilon]$ for an s-sparse $\vec{a}$ over $\mathbb{R}^d$. By writing this latter class as a union over the $2^{\binom{n}{k}}$ k-CNFs and $\binom{d}{s}$ coordinate subsets of size s, we find that it has VC-dimension $O(s \log s + s \log(d/s) + n^k) = O(\log d + n^k)$ for constant s.[1]

An optimal bound for sample complexity in terms of VC-dimension was recently obtained by Hanneke [11] (superseding the earlier bounds, e.g., by Vapnik [28] and Blumer et al. [4], although these would suffice for us, too): in this case, given

$$m = O\left(\frac{1}{\mu\gamma}\left(s \log s + s \log d + n^k + \log \frac{1}{\delta}\right)\right)$$

examples, if $[|\langle(\vec{a}',-1),(\vec{y},z)\rangle| \leq \varepsilon] \vee \neg h'(\vec{x})$ is consistent with all of the examples, then with probability $1 - \delta/3$ over the examples,

$$\Pr_{(x,y,z)\in D}[(|\langle(\vec{a}',-1),(\vec{y},z)\rangle| \leq \varepsilon) \vee \neg h'(\vec{x})] \geq 1 - \mu\gamma/2$$

or, equivalently,

$$\Pr_{(x,y,z)\in D}[(|\langle(\vec{a}',-1),(\vec{y},z)\rangle| > \varepsilon) \wedge h'(\vec{x})] \leq \mu\gamma/2.$$

Now, since for $m \geq \frac{4}{\mu\gamma} \ln \frac{3}{\delta}$, with probability $1 - \delta/3$,

$$\Pr_{(x,y,z)\in D}[h'(\vec{x})] \geq \frac{1-\gamma/2}{1+\gamma/2}\mu \geq (1-\gamma)\mu,$$

we find that for our choice of $\vec{a}'$ and h',

$$\Pr_{(x,y,z)\in D}\left[|\langle\vec{a}',\vec{y}\rangle - z| > \varepsilon \middle| h'(\vec{x})\right] \leq \frac{\gamma}{2}\frac{1+\gamma/2}{1-\gamma/2}$$

and so, $\Pr_{(x,y,z)\in D}\left[|\langle\vec{a}',\vec{y}\rangle - z| \leq \varepsilon \middle| h'(\vec{x})\right] \geq 1 - \gamma$ since $\gamma \leq 1/2$

as needed. ◀

Although it was not a focus of the analysis, we remark that the multiplicative Chernoff bound also guarantees that if *no* k-DNF event of probability greater than $(1-\gamma)\mu$ has a linear rule that is γ-close to having ϵ sup norm, then the algorithm is guaranteed to output INFEASIBLE with probability $1 - \delta$: the k-DNFs of probability less than $(1-\gamma)\mu$ fail the final test, and for the rest, the standard VC-dimension sample complexity analysis guarantees that we catch a point with error greater than the sup norm bound of ϵ in the sample (and so rule out the k-DNF during the Elimination algorithm). It follows that the algorithm is easily modified to solve a few natural variants of our problem. Suppose we return the list of all

[1] Exercises in Anthony and Biggs [2, Chapter 8, Exercise 6] and Mohri et al. [19, Exercise 3.15] on the growth function for ANDs/ORs of two distinct concept classes also yield this easily.

Algorithm 2: Dense Expected-error Regression Pigeonhole (DERP)

input : Examples $(\vec{x}^{(1)}, \vec{y}^{(1)}, z^{(1)}), \ldots, (\vec{x}^{(m)}, \vec{y}^{(m)}, z^{(m)})$, target fit ϵ.
output : A k-DNF over $x_1, \ldots, x_n$ and linear predictor over $y_1, \ldots, y_d$.
begin
  Initialize $c = \perp$, $\mu^* = 0$.
  forall *Terms T of size k over $x_1, \ldots, x_n$* **do**
    Put $S(T) = \{j : T(\vec{x}^{(j)}) = 1\}$.
    Let $\vec{a}$ minimize the squared-error on $(\vec{y}^{(j)}, z^{(j)})$ over $j \in S(T)$ subject to $\|\vec{a}\|_2 \leq B$.
    if $\frac{1}{m}\sum_{j \in S(T)}(\langle \vec{a}, \vec{y}^{(j)} \rangle - z^{(j)})^2 \leq 4\mu\epsilon$ *and* $|S(T)| \geq \mu^* m$ **then**
      Put $c = T$ and $\mu^* = |S(T)|/m$.
    end
  end
  return *c and $\vec{a}$*
end

coefficient vectors and k-DNFs that would pass our termination condition. We then obtain a list that contains all events that have probability μ (and only those that have probability at least $(1-\gamma)\mu$) for which the conditional distribution is γ-close to one where the linear rule has ϵ sup norm. Or, suppose we return the pair for which the k-DNF empirically satisfies the most examples. We then return a k-DNF that is within a $1-\gamma$ factor of having the largest probability (provided that this is at least μ) among those with a suitable linear rule.

4 Towards conditional dense, expected-error linear regression

While sparsity is a highly desirable feature to have of a linear regression fit, it may be the case that solutions are often not so sparse that Algorithm 1 is truly efficient. Moreover, we may also wish for an algorithm that handles an *expected error* variant of the regression task; the sup norm is particularly sensitive to noise or outliers, and thus is usually not a particularly desirable norm to use on real data. Our technique certainly does not address either of these concerns. The simple Algorithm 2 illustrates the best technique we currently have for either dense regression or expected error regression.

▶ **Theorem 9.** *Algorithm 2 solves the conditional ℓ_2-linear regression task: given access to a joint distribution D over $\vec{x} \in \{0,1\}^n$, $\vec{y} \in \mathbb{R}^d$ with $\|\vec{y}\|_2 \leq B$, and $z \in [-B, B]$ such that there is a k-DNF c and $\vec{a} \in \mathbb{R}^d$ with $\|\vec{a}\|_2 \leq B$ such that*

$$\mathop{\mathbb{E}}_{(x,y,z)\in D}\left[(\langle \vec{a}, \vec{y}\rangle - z)^2 \middle| c(\vec{x}) = 1\right] \leq \epsilon \qquad \text{and} \qquad 2\mu \geq \Pr_{(x,y,z)\in D}[c(\vec{x}) = 1] \geq \mu$$

and given B, k, ϵ, μ, and $\delta \in (0,1)$, using

$$m = O\left(\frac{B^8 n^k}{\mu\epsilon}\left(k\log n + \log\frac{1}{\delta}\right)\right)$$

examples from D, for any constant k, Algorithm 2 runs in polynomial time and finds a $\vec{a}'$ and k-DNF c' such that with probability $1-\delta$,

$$\mathop{\mathbb{E}}_{(x,y,z)\in D}\left[(\langle \vec{a}', \vec{y}\rangle - z)^2 \middle| c'(\vec{x}) = 1\right] \leq O(n^k\epsilon) \qquad \text{and} \qquad \Pr_{(x,y,z)\in D}[c'(\vec{x}) = 1] \geq \Omega(\mu/n^k).$$

Note that we can find such an estimate for μ by binary search.

Proof. We first observe that in particular, since for any T the objective function

$$\sum_{j \in S(T)} (\langle \vec{a}, \vec{y}^{(j)} \rangle - z^{(j)})^2$$

is convex, as is the set of $\vec{a}$ of ℓ_2-norm at most B, the main step of the algorithm is a convex optimization problem that can be solved in polynomial time, for example by gradient descent (see, e.g., [25, Chapter 14]). Thus, the algorithm can be implemented in polynomial time as claimed.

We next turn to correctness. Let c^* be the k-DNF promised by the theorem statement. By the pigeonhole principle, there must be some term T^* of c^* such that $\Pr[T^*(\vec{x}) = 1] \geq \mu / \binom{2n}{k}$. Observe that for the rule $\vec{a}^*$ promised to exist,

$$\begin{aligned}\mathop{\mathbb{E}}_{D}\left[(\langle \vec{a}^*, \vec{y} \rangle - z)^2 \middle| T^*(\vec{x}) = 1\right] \Pr_D[T^*(\vec{x}) = 1] &\leq \mathop{\mathbb{E}}_{D}\left[(\langle \vec{a}^*, \vec{y} \rangle - z)^2 \middle| c^*(\vec{x}) = 1\right] \Pr_D[c^*(\vec{x}) = 1] \\ &\leq \epsilon \cdot 2\mu.\end{aligned}$$

For a suitable choice of leading constant in the number of examples, a (multiplicative) Chernoff bound yields that with probability $1 - \delta/4$, at least $m \Pr[T^*(\vec{x}) = 1]/2$ examples satisfy T^* and noting that $(\langle \vec{a}^*, \vec{y} \rangle - z)^2 \in [0, 2B^4]$, with probability $1 - \delta/4$,

$$\frac{1}{m} \sum_{j=1}^{m} (\langle \vec{a}^*, \vec{y} \rangle - z)^2 T^*(\vec{x}) \leq 4\mu\epsilon$$

Thus, the $\vec{a}'$ minimizing the squared error on the set of examples also achieves

$$\frac{1}{m} \sum_{j : T^*(\vec{x}^{(j)}) = 1} (\langle \vec{a}', \vec{y}^{(j)} \rangle - z^{(j)})^2 \leq 4\mu\epsilon$$

as needed, so with probability $1 - \delta/2$, at least T^* is considered for c and the algorithm produces some c and $\vec{a}$ as output.

To see that any such T and $\vec{a}$ is satisfactory, we first note that any T we produce as output must satisfy at least as many examples as T^* by construction, so T must satisfy at least

$$\Pr_D[T^*(\vec{x}) = 1] m/2 \geq \Omega\left(\frac{B^8}{\epsilon}\left(k \log n + \log \frac{1}{\delta}\right)\right)$$

examples. In particular, this is at least $\mu m / 2\binom{2n}{k}$ examples, and a Chernoff bound guarantees that for suitable constants, with probability $1 - \delta/4\binom{2n}{k}$, no T with $\Pr_D[T(x) = 1] < \mu/4\binom{2n}{k}$ satisfies so many examples. Next, simply note that if for the best a for T with $\|\vec{a}\|_2 \leq B$, $\mathbb{E}_D\left[(\langle \vec{a}, \vec{y} \rangle - z)^2 | T(\vec{x}) = 1\right] \Pr[T(x) = 1] > 8\mu\epsilon$, then since $\|\vec{y}\|_2 \leq B$, $z^2 \leq B^2$, and the loss function is B-Lipschitz on this domain, a Rademacher bound (see, for example, [25, Theorem 26.12]) guarantees that with probability $1 - \delta/4\binom{2n}{k}$, for any such $\vec{a}$,

$$\frac{1}{m} \sum_{j : T(\vec{x}^{(j)}) = 1} (\langle \vec{a}, \vec{y}^{(j)} \rangle - z^{(j)})^2 > 4\mu\epsilon$$

and T will not be considered. A union bound over both events for all such T establishes that any T that is returned has, with probability $1 - \delta/2$, both

$$\Pr_D[T(\vec{x}) = 1] \geq \frac{\mu}{4\binom{2n}{k}} \text{ and } \mathop{\mathbb{E}}_{D}\left[(\langle \vec{a}, \vec{y} \rangle - z)^2 \middle| T(\vec{x}) = 1\right] \Pr_D[T(\vec{x}) = 1] \leq 8\mu\epsilon$$

and thus is as needed. Therefore, overall, with probability $1 - \delta$, the algorithm considers at least T^* as a candidate to output, and outputs a suitable term T and vector $\vec{a}$. ◀

5 Discussion and future directions

The main defect of Algorithm 2 is that in general it only recovers a condition with a $\Omega(1/n^k)$-fraction of the possible probability mass of the best k-DNF condition. This is in stark contrast to both Algorithm 1 and all of the earlier positive results for condition identification [15], in which we find a condition with probability at least a $(1-\gamma)$-fraction of that of the best condition, for any γ we choose. Indeed, we are most interested in the case where the probability of this event is relatively small and thus a $1/n^k$-fraction is extremely small. The main challenge here is to develop an algorithm for the dense and/or expected-error regression problem that similarly identifies a condition with probability that is a $(1-\gamma)$-fraction of that of the best condition.

Of course, the $O(n^k)$ blow-up in the expected error is also undesirable, but as indicated by Theorem 7, this is the same difficulty encountered in *agnostic learning*. Naturally, minimizing the amount by which constraints are violated is generally a harder problem than finding a solution to a system of constraints, and this is reflected in the quality of results that have been obtained. The results for such agnostic condition identification of k-DNFs in the previous work by Juba [15] suffers a similar blow-up in the error, which was recently improved to $\tilde{O}(\sqrt{n^k})$ by Zhang et al. [29]. The state-of-the-art algorithms for agnostic supervised learning for disjunctive classifiers by Awasthi et al. [3] suffer a similar blow-up of a $n^{k/3+o(1)}$-factor, and yet even for the harder problem of agnostic learning of linear threshold functions, only a sub-polynomial approximation factor is known to be necessary [7]. The question of what approximation factor is necessary is similarly wide open for our problem. We note briefly that a variant of Algorithm 2 in which we seek $\vec{a}$ satisfying $|\langle \vec{a}, \vec{y}^{(j)} \rangle - z^{(j)}| \leq \epsilon$ for all j satisfying a candidate term T solves the sup norm variant of Definition 2 for dense regression, and *does not* suffer this increase of the error. Of course, as mentioned earlier, one typically wishes to solve ℓ_1 or ℓ_2-norm regression, as these are much better behaved.

It is also natural to ask if instead of constant sparsity (as used here), we could simply bound the ℓ_1-norm of the coefficient vector, as in LASSO [27]. It's well known that this tends to produce sparse solutions (with ℓ_2-regression), without necessarily demanding an a priori fixed constant bound on the sparsity. Again, our technique does not achieve this.

Looking towards developing a better algorithm and solving further, related tasks, we note that a common strategy seems to be emerging. Our first algorithm, for the sup norm (Algorithm 1) operated by first generating a list of possible coefficient vectors for the regression fit, and then learning a condition that captures the various candidates in the list. This strategy is similar to the list-learning model independently introduced by Charikar et al. [6] for solving a variety of statistical problems when seeking to capture only a minority fraction of the data. Of course, in our work we ultimately sought to find a condition to single out one member of the list, rather than producing the entire list as output. Also, at a technical level, while their work applies to a much, much broader variety of problems, their technique also suffers an increase in the losses that grows with $1/\sqrt{\mu}$. (The relatively trivial Algorithm 2, as discussed above, suffers a n^k-factor increase, but has no dependence on μ, which naturally may be more or less desirable depending on the setting.) These technical differences aside, we believe that both of these works suggest that a relatively broad family of problems involving statistics of minority sub-populations may be tackled by variants of the list-learning approach. Indeed, we observe that the problem is essentially similar to that tacked by list-decoding in coding theory (e.g., see Sudan [26] for an overview): although the amount of agreement with the data may be too small to uniquely determine a "best" hypothesis, it may be possible to output a small list containing all possible hypotheses. Although most of the work in coding

theory is focused on finite characteristic (in contrast to most problems we would seek to solve in data analysis), it may be informative.

One immediate family of questions to be addressed is, which *conditional* variants of the standard supervised learning tasks can be solved efficiently? In particular, when can such tasks solved without suffering an increase in the loss that depends polynomially on $1/\mu$? For example, for which families of Boolean classification tasks do such algorithms exist? We know that the conditions (essentially) must be described by k-DNFs, but this seems to tell us nothing about which rules we can fit on such conditional distributions.

Acknowledgements. I thank Madhu Sudan for originally suggesting the joint problem of learning under conditional distributions. I also thank Ben Moseley for many helpful discussions about these problems. Finally, I thank the reviewers for their comments and suggestions.

References

1 Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, chapter 12, pages 307–328. MIT Press, Cambridge, MA, 1996.

2 Martin Anthony and Norman Biggs. *Computational Learning Theory*. Number 30 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, New York, NY, 1992.

3 Pranjal Awasthi, Avrim Blum, and Or Sheffet. Improved guarantees for agnostic learning of disjunctions. In *Proc. 23rd COLT*, pages 359–367, 2010.

4 Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989. doi:10.1145/76359.76371.

5 Nader H. Bshouty and Lynn Burroughs. Maximizing agreements with one-sided error with applications to heuristic learning. *Machine Learning*, 59(1–2):99–123, 2005. doi:10.1007/s10994-005-0464-5.

6 Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning with untrusted data. arXiv:1611.02315, 2016.

7 Amit Daniely. Complexity theoretic limitations on learning halfspaces. In *Proc. 48th STOC*, pages 105–117, 2016. doi:10.1145/2897518.2897520.

8 Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limtations on learning DNF's. In *Proc. 29th COLT*, volume 49 of *JMLR Workshops and Conference Proceedings*, pages 815–830, 2016.

9 Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. doi:10.1145/358669.358692.

10 Jerome H. Friedman and Nicholas I. Fisher. Bump hunting in high-dimensional data. *Statistics and Computing*, 9(2):123–143, 1999. doi:10.1023/A:1008894516817.

11 Steve Hanneke. The optimal sample complexity of PAC learning. *JMLR*, 17(38):1–15, 2016.

12 Moritz Hardt and Ankur Moitra. Algorithms and hardness for robust subspace recovery. In *Proc. 26th COLT*, volume 30 of *JMLR Workshops and Conference Proceedings*, pages 354–375, 2013.

13 Peter J. Huber. *Robust Statistics*. John Wiley & Sons, New York, NY, 1981.

14 Jiming Jiang. *Linear and Generalized Linear Mixed Models and Their Applications*. Springer, Berlin, 2007.

15 Brendan Juba. Learning abductive reasoning using random examples. In *Proc. 30th AAAI*, pages 999–1007, 2016.

16 Adam Tauman Kalai, Varun Kanade, and Yishay Mansour. Reliable agnostic learning. *JCSS*, 78:1481–1495, 2012. doi:10.1016/j.jcss.2011.12.026.

17 Varun Kanade and Justin Thaler. Distribution-independent reliable learning. In *Proc. 27th COLT*, volume 35 of *JMLR Workshops and Conference Proceedings*, pages 3–24, 2014.

18 Charles E. McCulloch and Shayle R. Searle. *Generalized, Linear, and Mixed Models*. John Wiley & Sons, New York, NY, 2001.

19 Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, Cambridge, MA, 2012.

20 B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24(2):227–234, 1995. doi:10.1137/S0097539792240406.

21 Leonard Pitt and Leslie G. Valiant. Computational limitations on learning from examples. *J. ACM*, 35(4):965–984, 1988. doi:10.1145/48014.63140.

22 Avi Rosenfeld, David G. Graham, Rifat Hamoudi, Rommell Butawan, Victor Eneh, Saif Kahn, Haroon Miah, Mahesan Niranjan, and Laurence B. Lovat. MIAT: A novel attribute selection approach to better predict upper gastrointestinal cancer. In *Proc. IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–7, 2015. doi:10.1109/DSAA.2015.7344866.

23 Peter J. Rousseeuw and Annick M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, New York, NY, 1987.

24 Alexander Schrijver. *Theory of Linear and Integer Programming*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, 1986.

25 Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, 2014.

26 Madhu Sudan. List decoding: Algorithms and applications. In J. van Leeuwen, O. Watanabe, M. Hagiya, P.D. Mosses, and T. Ito, editors, *IFIP International Converence on Theoretical Computer Science*, volume 1872 of *LNCS*, pages 25–41. Springer, 2000. doi:10.1007/3-540-44929-9_3.

27 Robert Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal Statist. Soc. B*, 58(1):267–288, 1996.

28 Vladimir Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer, New York, NY, 1982.

29 Mengxue Zhang, Tushar Mathew, and Brendan Juba. An improved algorithm for learning to perform exception-tolerant abduction. To appear in 31st AAAI, 2017.

30 Yuchen Zhang, Martin J. Wainwright, and Michael I. Jordan. Lower bounds on the performance of polynomial-time algorithms for sparse linear regression. In *Proc. 27th COLT*, volume 35 of *JMLR Workshops and Conference Proceedings*, pages 921–948, 2014.

Rigorous Rg Algorithms and Area Laws for Low Energy Eigenstates In 1D

Itai Arad[*1], **Zeph Landau**[†2], **Umesh Vazirani**[‡3], **and Thomas Vidick**[§4]

1 **Centre for Quantum Technologies (CQT), National University of Singapore, Singapore**
arad.itai@fastmail.com
2 **Electrical Engineering and Computer Sciences, University of California, Berkeley, USA**
zeph.landau@gmail.com
3 **Electrical Engineering and Computer Sciences, University of California, Berkeley, USA**
vazirani@eecs.berkeley.edu
4 **Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, USA**
vidick@cms.caltech.edu

Abstract

One of the central challenges in the study of quantum many-body systems is the complexity of simulating them on a classical computer. A recent advance [8] gave a polynomial time algorithm to compute a succinct classical description for unique ground states of gapped 1D quantum systems. Despite this progress many questions remained unresolved, including whether there exist rigorous efficient algorithms when the ground space is degenerate (and poly(n) dimensional), or for the poly(n) lowest energy states for 1D systems, or even whether such states admit succinct classical descriptions or area laws.

In this paper we give a new algorithm for finding low energy states for 1D systems, based on a rigorously justified renormalization group (RG)-type transformation. In the process we resolve some of the aforementioned open questions, including giving a polynomial time algorithm for poly(n) degenerate ground spaces and an $n^{O(\log n)}$ algorithm for the poly(n) lowest energy states for 1D systems (under a mild density condition). We note that for these classes of systems the existence of a succinct classical description and area laws were not rigorously proved before this work. The algorithms are natural and efficient, and for the case of finding unique ground states for frustration-free Hamiltonians the running time is $\tilde{O}(nM(n))$, where $M(n)$ is the time required to multiply two $n \times n$ matrices.

1998 ACM Subject Classification F.2.1 Numerical Algorithms and Problems, J.2 Physical Sciences and Engineering

Keywords and phrases Hamiltonian complexity, area law, gapped ground states, algorithm

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.46

* I. Arad's research was partially performed at the Centre for Quantum Technologies, funded by the Singapore Ministry of Education and the National Research Foundation, also through the Tier 3 Grant random numbers from quantum processes.
† The author acknowledges support by ARO Grant W911NF-12-1-0541, NSF Grant CCF-1410022 and Templeton Foundation Grant 52536.
‡ The author acknowledges support by ARO Grant W911NF-12-1-0541, NSF Grant CCF-1410022 and Templeton Foundation Grant 52536.
§ T. Vidick was partially supported by the IQIM, an NSF Physics FrontiersCenter (NFS Grant PHY-1125565) with support of the Gordon and Betty Moore Foundation (GBMF-12500028).

© Itai Arad, Zeph Landau, Umesh Vazirani, and Thomas Vidick;
licensed under Creative Commons License CC-BY

1 Introduction

One of the central challenges in the study of quantum many-body systems is the classical complexity of finding low energy states of local Hamiltonians. This problem is the quantum analog of constraint satisfaction problems, and is known to be QMA-complete, even for one-dimensional Hamiltonians [1]. This means that we do not even expect that there is a sub-exponential size classical description of the ground state that allows efficient computation of local observables such as the energy. In sharp contrast, the widely used heuristic density matrix renormalization group (DMRG) algorithm invented two decades ago [14] has been remarkably successful in practice on one-dimensional (1D) problems. A recent advance [8] resolved this seeming contradiction by giving a polynomial time algorithm to actually compute a succinct classical description for unique ground states of quantum systems coming from *gapped* 1D Hamiltonians.

In this paper we give a fundamentally different algorithm that applies to a significantly larger class of 1D Hamiltonians, namely:

1. *Hamiltonians with a degenerate gapped ground space:* H has smallest eigenvalue ε_0 with associated eigenspace of dimension $r = \text{poly}(n)$, and second smallest eigenvalue ε_1 such that $\varepsilon_1 - \varepsilon_0 \geq \gamma$.
2. *Gapless Hamiltonians with a low density of low-energy states:* The dimension of the space of all eigenvectors of H with eigenvalue in the range $[\varepsilon_0, \varepsilon_0 + \eta]$, for some constant $\eta > 0$, is $r = \text{poly}(n)$.

For both classes of Hamiltonians, our results show the existence of succinct representations in the form of matrix product states (MPS) for a basis for (a good approximation to) the ground space (resp. low energy subspace) of the Hamiltonian. The bond dimension of the MPS is polynomial in r and n and exponential in μ^{-1} (respectively η^{-1} in case 2). Furthermore the algorithms return these MPS descriptions in polynomial time for the first case and quasi-polynomial time in the second. For the special case of finding unique ground states for frustration-free Hamiltonians the algorithm is particularly efficient, with a running time of $\tilde{O}(nM(n))$, where $M(n)$ is the time required to multiply two $n \times n$ matrices.

Our results should be understood in the context of a substantial body of prior work studying ground state entanglement in 1D systems. Central to this work is the so-called *area law for entanglement entropy* — a conjecture which states that in ground states of gapped local Hamiltonians the entanglement entropy of a region scales as its surface area, rather than its volume. A landmark result by Hastings [5] proved this conjecture for gapped 1D systems with unique ground state, and a sequence of follow up results substantially strengthened the bounds (see, for example, the review article Ref. [4]). However, the techniques for these results break down for low energy and degenerate ground states, and few results were known for these questions: Chubb and Flammia [3] extended the approach from Ref. [8] and subsequent improvements from Ref. [6] to establish an efficient algorithm (and area law) for gapped Hamiltonians with a constant degeneracy in the ground space. Masanes [9] proves an area law with logarithmic correction under a strong assumption on the density of states, together with an additional assumption on the exponential decay of correlations in the ground state.

Our algorithm hinges on novel ideas that can be viewed as giving a rigorous underpinning to the well known Renormalization Group (RG) formalism within condensed matter physics [15]. This formalism occupies a central place in many-body physics and provides a sweeping computational approach to the "physically relevant corner of Hilbert space" by suggesting that such states can be coarse-grained at different levels of granularity, or length scales,

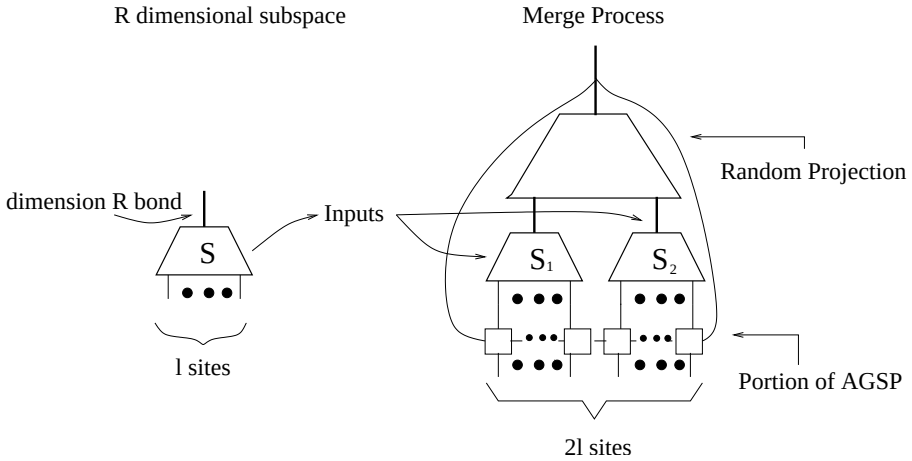
thereby iteratively eliminating the irrelevant degrees of freedom. There are recent suggestions that RG may be intimately related to feature extraction and the success of deep nets [10].

The new algorithm relies on three innovations. The first innovation starts with the idea that if our goal is to approximate a subspace T (of low energy states, say) on n qubits, the algorithm can make progress by locally maintaining a small dimensional subspace S on k particles, with the property that T is close to $S \otimes X$ where X is the entire Hilbert space on the remaining $n-k$ particles. A major challenge here is measuring the quality of this partial solution. This is accomplished by a suitable generalization of the definition of a viable set introduced in [8] to the setting of a target subspace T, and is one of the conceptual contributions of this paper.

As the number of qubits k on which the partial solution lies is increased, the fundamental challenge is to keep the dimension of the subspace S, which will otherwise naturally grow exponentially, polynomial, while maintaining the quality of the solution which will otherwise decay. To address this, we design a two step process: we deal with the dimension problem by projecting onto a small random subspace. Of course this degrades the quality i.e. blows up the error. So in turn, we control the error by the application of an AGSP, an operator with small entanglement rank which preserves low-energy states while reducing the norm of high energy states. The success of this two-step process depends upon the construction of a suitable class of AGSPs — uniform AGSPs — that ensures a favorable tradeoff between dimension and error between random projection and AGSP. The two-step process may be visualized as a (two stroke) pump: the first step (random projection) reduces the dimension, but leaves the dimension-error tradeoff essentially unchanged. The second step (AGSP) improves the dimension-error tradeoff sufficiently, so that the two steps together restore both dimension and error to their values at the start of the iteration.

The third innovation is to design an algorithm whose analysis does not rely on an area law: instead of processing one particle per iteration, it acts in parallel on all sites at each iteration, merging the results along a binary tree. Consequently, the algorithm uses only $O(\log(n))$ iterations. This is important because besides the number of quantum states stored (the dimension of the subspace S), the running time of the algorithm is also governed by the description complexity (bond dimension) of these states. The bond dimension grows exponentially with the number of iterations, yielding an $n^{O(\log n)}$ algorithm.

A tensor network picture of the resulting process, called Merge, is provided in the figure below.[1] Beginning with inputs representing subspaces of ℓ qubits shown on the left, the Merge process (shown on the right) outputs a representation of a small subspace on 2ℓ qubits.



[1] We are grateful to Christopher T. Chubb for originally suggesting these pictures to us.

The algorithm iteratively applies the Merge process. The overall construction results in a partial isometry that is reminiscent of a MERA [12, 13], and it should be fruitful to compare and contrast them as well as to standard RG. In particular, both our construction and RG build subspaces in a binary tree fashion. However, whereas RG can be realized as a tensor network on a binary tree (where each node represents the partial isometry associated with selecting only a small portion of the previous space), the use of the AGSP in our construction allows for selection of the small subspace outside the tensor product of the previous two spaces.

This hierarchical merge process establishes a new operational description of the entanglement structure of the low energy states of local Hamiltonians in 1D. It allows us to prove an area law for poly(n) degenerate ground spaces and (up to log correction) for low energy states, and is the basis of the rigorously justified RG transformation.

Our new algorithms could potentially be made very efficient. The main bottlenecks are the complexity of the AGSP and the MPS bond dimension that must be maintained. In the case of a frustration-free Hamiltonian with unique ground state we obtain a running time of $O(2^{O(1/\gamma^2)}n^{1+o(1)}\mathrm{M}(n))$, where $\mathrm{M}(n)$ is matrix multiplication time. This has an exponentially better scaling in terms of the spectral gap γ (due to avoidance of the ε-net argument) and saves a factor of $n/\log n$ (due to the logarithmic, instead of linear, number of iterations) as compared to an algorithm for the same problem considered in [6]. We speculate that it might further be possible to limit the bond dimension of all MPS considered to $n^{o(1)}$ (instead of $n^{1+o(1)}$ currently), which, if true, would imply a nearly-linear time $O(n^{1+o(1)})$ algorithm.

Organization of the paper

We start with some preliminaries and notation in Section 2. Section 3 introduces our main tool, the Merge process, and employs it to derive an area law for 1D gapped Hamiltonian with polynomial degeneracy in the ground space. In Section 4 we build on the approach to develop an efficient algorithm for the same systems. The constructions of AGSP that underlie our results, as well as the algorithm for the case of gapless Hamiltonians with a low density of low-energy states, are described in the full version of this paper [2].

2 Preliminaries and Notation

We begin by describing the basic setup for our results.

▶ **Definition 1.** Let $H = \sum_{i=1}^{n} h_i$ be a local Hamiltonian acting on the Hilbert space

$$\mathcal{H} = \mathbb{C}^d \otimes \mathbb{C}^d \otimes \cdots \otimes \mathbb{C}^d \simeq (\mathbb{C}^d)^{\otimes n}$$

associated with a 1D chain of n qudits, each of local dimension d. Each h_i is assumed to be a non-negative operator with norm at most 1 acting on the i-th and $(i+1)$-st qudits. We denote by $\varepsilon_0 \geq 0$ the smallest eigenvalue (ground energy) of H, and consider the following assumptions:

(FF) Frustration-Free: H is frustration-free ($\varepsilon_0 = 0$) with a unique ground state $|\Gamma\rangle$ and a spectral gap $\gamma > 0$ above the ground state. In this case we let $T = \mathrm{Span}\{|\Gamma\rangle\}$ denote the one-dimensional ground space of H.

(DG) Degenerate Gapped: H has a degenerate ground space T of dimension $r = \mathrm{poly}(n)$, along with a spectral gap $\gamma > 0$ above the ground space.

For $A \subseteq \{1, \cdots, n\}$ we denote the Hilbert associated with the qudits in A by $\mathcal{H}_A$, e.g. $\mathcal{H}_{[1,3]} = \mathbb{C}^d \otimes \mathbb{C}^d \otimes \mathbb{C}^d$ corresponds to the first three qudits. Separately, for any operator (Hamiltonian) H, $H_{[a,b]}$ will denote the subspace spanned by the eigenvectors of H with eigenvalues in the interval $[a, b]$. For a set S of vectors we denote by P_S the orthogonal projection onto the span of S and refer to $\dim(\mathrm{Span}(S))$ as the *size* of S, denoted $|S|$. We often identify sets of vectors with the vector space they span.

We use standard $O(\cdot), o(\cdot), \Omega(\cdot), \omega(\cdot)$ and $\Theta(\cdot)$ notation. The use of a tilde, such as $\tilde{O}(\cdot)$, will indicate a polylogarithmic overhead, i.e. $\tilde{O}(f) = O(f \operatorname{poly} \log f)$. We use $f = \mathrm{poly}(n)$ to mean that there is a fixed polynomial p such that $f(n) \leq p(n)$ for all n.

3 Viable sets, the merge process, and area laws

Recall that our goal is to formalize and analyze an RG-like transformation in the spirit of the following claim, which for ease of explanation we state for the gapped degenerate case:

▶ **Proposition 2.** *Let H be a local Hamiltonian satisfying Assumption (DG) (q.v. Definition 1).*

1. *For every length scale ℓ and contiguous block A of ℓ qudits, there is a subspace $S \subseteq \mathcal{H}_A$ of dimension $q = r^{1+o(1)} e^{\tilde{O}\left(\frac{1}{\gamma} \log^3 d\right)}$ such that S* approximates *the ground space T of H, in the following sense: every state in T has large overlap with a state whose reduced density matrix on $\mathcal{H}_A$ is supported on S.*
2. *Suppose given two subspaces $S_1 \subseteq \mathcal{H}_1$ and $S_2 \subseteq \mathcal{H}_2$ on adjacent blocks of ℓ qudits each, such that each of S_1, S_2 has dimension at most q and approximates T. Then it is possible to generate a subspace $S \subseteq \mathcal{H}_1 \otimes \mathcal{H}_2$ of the composite system that has the* same *dimension q and approximates T to the* same *extent as S_1, S_2.*

The key feature of the second item in the proposition is that the dimension of the merged set S has not increased: the set has the same size as S_1, S_2 separately, and yet it combines all the information each of these sets holds about the restriction of the ground space T to $\mathcal{H}_1$ and $\mathcal{H}_2$ respectively.

As we will see, the first item in the proposition leads naturally to an area law and succinct MPS representations for good approximations to states in T. The proof of the first item will be obtained by iteratively performing the merging procedure described in the second item. With additional work the merging procedure can be made efficient, leading to an efficient algorithm for computing these succinct representations.

3.1 Viable sets

We formalize the notion of a subspace *approximating* another as follows.

▶ **Definition 3.** A subspace T is *δ-close* to a subspace T' if

$$P_{T'} P_T P_{T'} \geq (1 - \delta) P_{T'}.$$

We say that T and T' are *mutually δ-close* if each is δ-close to the other, and denote by $\angle_m(T, T')$ the smallest δ such that T, T' are mutually δ-close.

Geometrically,

$$\angle_m(T, T') = 1 - \min_{\substack{x \in T \\ \|x\|=1}} \max_{\substack{x' \in T' \\ \|x'\|=1}} |x \cdot x'|^2$$

is the squared sine of the largest principal angle between the subspaces T and T' (where the cosines of the principal angles are given by the singular values of $P_T P'_T$); in particular the statement that T is δ-close to T' is equivalent to the fact that for every $|\psi\rangle \in T'$ there exists $|\phi\rangle \in T$ such that $|\langle\psi|\phi\rangle|^2 \geq 1-\delta$. Note that mutually close subspaces always have the same dimension.

With a view towards working with subsystems, we extend the notion of closeness to capture approximation by subspaces defined only on one half of a factored Hilbert space $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$.

▶ **Definition 4.** Given a subspace $T \subseteq \mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$, a subspace $S \subseteq \mathcal{H}_A$ is *δ-viable* for T if

$$P_T P_{S_{ext}} P_T \geq (1-\delta) P_T, \tag{1}$$

where $S_{ext} := S \otimes \mathcal{H}_B$.

This definition generalizes the definition of a viable set from Ref. [8], which was specialized to the case where T is a one-dimensional subspace containing a unique ground state. Informally it captures the notion that a reasonable approximation of T can be made using the subspace $S \otimes \mathcal{H}_B$. With the definition in place, we can make the statement of item 1 of Proposition 2 precise. For ease of exposition we relax the dependence of q on r to $\tilde{O}(r^2)$.

▶ **Theorem 5.** *Let H be a local Hamiltonian satisfying Assumption (DG). Then for any block $A \subseteq \{1,\ldots,n\}$ of $\ell \leq n$ qudits there exists a .015-viable set $S \subset \mathcal{H}_A$ for the ground space T of H of dimension at most $q = \tilde{O}(r^2) e^{\tilde{O}\left(\frac{1}{\gamma}\log^3 d\right)}$.*

We further note that the dependence of q on the dimension r can be improved to $\tilde{O}(r)$, using a direct "bootstrapping" argument that is slightly different from the more "algorithmic" argument that we give in Section 3.3. This improved bound and bootstrapping argument are presented in the full version [2].

While the notion of viable set is quite intuitive for small δ, our arguments will also involve viable sets with parameter δ close to 1, a regime where there is less intuition. A helpful interpretation of the definition is that it formalizes the fact that for a viable set S, the image of the unit ball of S_{ext} when projected to T contains the ball of radius $(1-\delta)$.

▶ **Lemma 6.** *If S is δ-viable for T then for every $|t\rangle \in T$ of unit norm, there exists an $|s\rangle \in S_{ext}$ such that $P_T|s\rangle = |t\rangle$ and $\||s\rangle\| \leq \frac{1}{1-\delta}$.*

The proof of this and the following two useful lemmas appear in the full version [2]. The first summarizes the effect of tensoring two viable sets supported on disjoint spaces.

▶ **Lemma 7.** *Suppose S_1, S_2 are δ_1-viable and δ_2-viable for T respectively, defined on disjoint sets of qudits. Then the set $S := S_1 \otimes S_2$ is $(\delta_1+\delta_2)$-viable for T.*

The second lemma shows that our notion of closeness can be chained together and is compatible with the notion of viable set:

▶ **Lemma 8.** *If T is δ-close to T' and T' is δ'-close to T'' then T is $2(\delta+\delta')$-close to T''. Consequently if S is δ-viable for T and T is δ'-close to T' then S is $2(\delta+\delta')$-viable for T'.*

3.2 The Merge Process

We are ready to outline the merging procedure referred to in item 2 of Proposition 2, which lies at the heart of our RG transformation. Assume we are given a decomposition $\mathcal{H} = \mathcal{H}_L \otimes (\mathcal{H}_1 \otimes \mathcal{H}_2) \otimes \mathcal{H}_R$ of the global Hilbert space. The merge process **Merge** takes as input two subsets $V_1 \subseteq \mathcal{H}_1$ and $V_2 \subseteq \mathcal{H}_2$ and returns a subset $V \subseteq \mathcal{H}_1 \otimes \mathcal{H}_2$. To do so, it requires two additional inputs: a finite set of operators $\{A_i\}_{i=1}^{D^2}$ each acting on $\mathcal{H}_1 \otimes \mathcal{H}_2$, along with a positive integer s. The procedure consists of the following three simple steps.

Merge$(V_1, V_2, \{A_i\}, s)$**:**
Step 1: Tensoring. Set $W = V_1 \otimes V_2$.
Step 2: Random Sampling. Let $W' \subseteq W$ be a random s-dimensional subspace of W.
Step 3: Error Reduction. Set $V = \mathrm{Span}(\cup_i A_i W')$.
Return V.

The effectiveness of **Merge** relies on the properties of the operators $\{A_i\}$, with a sufficiently good choice of these operators leading to a formalization of item 2. of Proposition 2. Suitable operators can be obtained from the decomposition of an *approximate ground state projection* (AGSP). The detailed construction is given in the full paper [2]; the following theorem summarizes the essential properties of the resulting $\{A_i\}$.

▶ **Theorem 9** (Existence of AGSP, (DG))**.** *Let H be a local Hamiltonian satisfying Assumption (DG), and $\mathcal{H} = \mathcal{H}_L \otimes \mathcal{H}_M \otimes \mathcal{H}_R$ a decomposition of the n-qudit space in three contiguous blocks. There exists a collection of D^2 operators $\{A_i\}_{i=1}^{D^2}$ acting on $\mathcal{H}_M$ along with a subspace $\tilde{T} \subseteq \mathcal{H}$ such that:*

- $\angle_m(T, \tilde{T}) \leq .005$,
- $D = e^{\tilde{O}\left(\frac{1}{\gamma}\log^3 d\right)}$,
- *There is $\Delta > 0$ such that $D^{12}\Delta \leq \frac{1}{2000}$ and whenever $S \subseteq \mathcal{H}_M$ is δ-viable for $\tilde{T}$ then $S' = \mathrm{Span}\{\cup_i A_i S\}$ is δ'-viable for $\tilde{T}$, with $\delta' = \frac{\Delta}{(1-\delta)^2}$.*

Given a finite collection of operators $\{A_i\}$ we denote by $\{A_i\}^k$ the set of all products of k of the A_i. The following theorem states the guarantees offered by the **Merge** process when initialized with operators $\{A_i\}$ satisfying the guarantees of Theorem 9.

▶ **Theorem 10.** *Let H be a local Hamiltonian satisfying Assumption (DG), and $\mathcal{H} = \mathcal{H}_L \otimes (\mathcal{H}_1 \otimes \mathcal{H}_2) \otimes \mathcal{H}_R$ a decomposition of the n-qudit space into contiguous blocks. Let $\{A_i\}$ and D be as in Theorem 9,*

$$s \geq 1600r(\log r + 1) \qquad \text{and} \qquad k = \frac{1}{2}\lceil \log_D(s) \rceil.$$

Let $V_1 \subseteq \mathcal{H}_1$ and $V_2 \subseteq \mathcal{H}_2$ be .015-viable subspaces for T of size $q = s^2$ each. Then with probability $1 - e^{-\Omega(s)}$ the space $V = \mathbf{Merge}(V_1, V_2, \{A_i\}^k, s)$ is .015-viable for T with $|V| \leq q$.

The proof below analyzes the effect of each of the three steps of the **Merge** process. The first creates the trivial subspace $V_1 \otimes V_2$, whose dimension $q^2 = \dim(V_1)\dim(V_2)$ is too large, and whose overlap with T is worse than desired by a factor of 2. The random sampling step roughly evenly trades off size for overlap: it picks a random s dimensional subspace for $s \ll q$, at the expense of making the overlap roughly $\frac{s}{q^2}$. Finally, the application of the AGSP (via the operators $\{A_i\}$) blows up the size from s to at most q, while increasing overlap to at least the original overlap of V_1 and V_2. This relies on the highly favorable D, Δ-tradeoff of the AGSP.

Proof. We analyze each of the three steps of the **Merge** process:

1. Tensoring. Applying Lemma 7 yields that the result of step 1, $W = V_1 \otimes V_2 \subseteq \mathcal{H}_1 \otimes \mathcal{H}_2$, is a .03 viable set for T of size q^2. Using the first condition from Theorem 9 and applying Lemma 8, W is .07-viable for $\tilde{T}$.

2. Random Sampling. We show that at the end of this step, with high probability W' is $(1-\alpha)$-viable for $\tilde{T}$ with $\alpha = (.8)s/q^2$. We accomplish this by establishing that with high probability $\|P_{W'_{ext}}|v\rangle\|^2 \geq \alpha$ for all states $|v\rangle \in \tilde{T}$, where $W'_{ext} = \mathcal{H}_L \otimes W' \otimes \mathcal{H}_R$ and $W_{ext} = \mathcal{H}_L \otimes W \otimes \mathcal{H}_R$.

Let $|v\rangle \in \tilde{T}$ have norm 1, and $|w\rangle = P_{W_{ext}}|v\rangle \in W_{ext}$. Using that W is .07-viable for $\tilde{T}$ it follows that $\||w\rangle\|^2 \geq .995$. Since $W'_{ext} \subseteq W_{ext}$, $P_{W'_{ext}}|v\rangle = P_{W'_{ext}}|w\rangle$. Applying a standard concentration argument it holds that $\|P_{W'_{ext}}|v\rangle\|^2 \geq (.9)(.995)\frac{s}{q^2}$ with probability at least $1 - q^2 e^{-s/400}$.

By a simple volume argument (see e.g. [11, Lemma 5.2]) there exists a ν-net for the Euclidean unit ball of $\tilde{T}$ consisting of at most $(1+\frac{2}{\nu})^r$ elements of $\tilde{T}$, where $\nu = \sqrt{(.1)(.9)(.995)\frac{s}{q^2}}$. Applying the preceding argument to each $|v\rangle$ in the net, a choice of s such that

$$\eta = \left(1 + \frac{2}{\nu}\right)^r q^2 e^{-s/400} < 1 \tag{2}$$

will guarantee that with probability at least $1-\eta$, $\|P_{W'_{ext}}|v\rangle\|^2 \geq (.9)(.995)\frac{s}{q^2}$ for all $|v\rangle$ in the net; hence $\|P_{W'_{ext}}|v\rangle\|^2 \geq (.99)(.9)(.995)\frac{s}{q^2} \geq .8\frac{s}{q^2}$ for all $|v\rangle \in \tilde{T}$ of unit norm. The equation (2) is satisfied with

$$s > 400\Big(2\log q + \frac{r}{2}\log\Big(1 + \sqrt{47\frac{q^2}{s}}\Big)\Big),$$

a condition verified by the choices of s and q made in the theorem.

Step 3: Error Reduction. Applying Theorem 9 k times in sequence, $V = \mathrm{Span}\{\{A_i\}^k \cdot W'\}$ is $\frac{\Delta^k}{(1-\delta)^2} = \frac{\Delta^k}{\alpha^2}$-viable for T' of size at most $D^{2k}s$. Our choice of k ensures $D^{2k}s = q$, and the relation between D and Δ implies that

$$\frac{\Delta^k}{\alpha^2} = \frac{s^6\Delta^k}{.64} = \frac{D^{12k}\Delta^k}{.64} \leq \frac{1}{(.64)}\frac{1}{2000} \leq .001.$$

Thus V is .001-viable for $\tilde{T}$, and by Lemma 8 it is $.012 < .015$-viable for T. ◀

3.3 Area law for degenerate Hamiltonians

In this section we first prove Theorem 5 establishing the claim made in the first item of Proposition 2. From the theorem we then deduce an area law for local Hamiltonians satisfying Assumption (DG) (degenerate ground space with a spectral gap; q.v. Definition 1).

Proof of Theorem 5. Consider a system A of $\ell \leq n$ consecutive qudits; for ease of notation we'll assume that ℓ is a power of 2 and A consists of the first ℓ qudits of the n-qudit chain on which H acts. The proof of the theorem is based on the following iterative procedure for constructing the .015-viable set claimed in the theorem. The procedure depends on a set of operators $\{A_i\}$ obtained from Theorem 9 for various decompositions of $\mathcal{H}$, and we let s and k be as in the theorem.

The proof of Theorem 5 follows by showing that with positive probability Procedure 1 returns a subspace $V_1^{\log \ell}$ that is .015-viable for T and such that $|V_1^{\log \ell}| = \tilde{O}(r^2)e^{\tilde{O}\left(\frac{1}{\gamma}\log^3 d\right)}$.

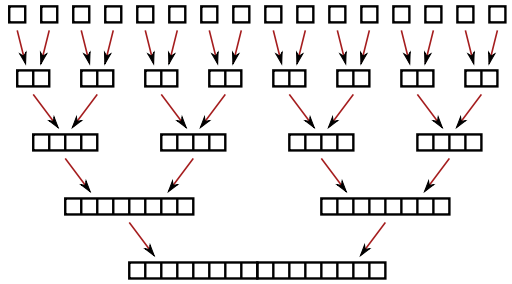


Figure 1 The parallel structure of Procedure 1. Each square represents a qudit, and successive viable sets are supported on neighboring groups of squares.

Procedure 1: Given a local Hamiltonian H satisfying(DG), returns a viable set for T supported on the first ℓ qudits.

Initialization: Set $V_j^0 = \mathcal{H}_j$ for $j = 1, 2, \ldots \ell$.

Iteration:

For $i = 1, \ldots, \log(\ell)$ do:

For all $j \in \{1, 2, \ldots \frac{\ell}{2^i}\}$, set

$$V_j^i = \textbf{Merge}(V_{2j-1}^{i-1}, V_{2j}^{i-1}, \{A_i\}^k, s) \subseteq \mathcal{H}_{[(j-1)2^i+1, j2^i]},$$

where $\{A_i\}$ are as in Theorem 9 for the decomposition

$$\mathcal{H} = \mathcal{H}_{[1,(j-1)2^i]} \otimes (\mathcal{H}_{[(j-1)2^i+1,(2j-1)2^{i-1}]} \otimes \mathcal{H}_{[(2j-1)2^{i-1}+1, j2^i]}) \otimes \mathcal{H}_{[j2^i+1,\ell]},$$

and s and k are as in Theorem 10.

Let $q = D^{2k}s$ be the size of the space output by **Merge**, and observe that by Theorem 9 it holds that $q = \tilde{O}(r^2)e^{\tilde{O}\left(\frac{1}{\gamma}\log^3 d\right)}$. We prove the result by induction, showing that V_j^i is .015-viable for T with $|V_j^i| \leq q$ for each i, j. The initialization step establishes this for $i = 0$ since each V_j^0 is 0-viable for T with $|V_j^0| = d$. The induction step is a direct consequence of Theorem 10, which establishes that at each iteration with probability $1 - e^{-\Omega(s)}$ the set $V_j^i = \textbf{Merge}(V_{2j-1}^{i-1}, V_{2j}^{i-1}, s, \{A_i\}^k)$ is .015-viable for T with $|V_j^i| \leq q$. Each merging operation succeeds with independent probability, therefore there is a positive probability that the procedure terminates with a .015-viable set $V_1^{\log \ell}$ for T. ◀

An area law for gapped Hamiltonians with a degenerate ground space follows readily from Theorem 5. Indeed, for any desired cut Theorem 5 establishes the existence of 0.015-viable sets of size at most q for the block of qudits on either side of the cut. As a consequence each element of the ground space T has a constant approximation by a state of Schmidt rank at most q. Applying a suitable AGSP to the tensor product of two such viable sets one can obtain a δ-viable set, for any desired δ, at a modest (depending on δ) increase in size. This kind of trade-off leads to a standard proof bounding both the Schmidt rank and the von Neumann entropy across the cut for any state in the ground space. We state the result here, referring to the full version [2] for the proof.

▶ **Corollary 11** (Area law for degenerate gapped Hamiltonians). *Let H be a local Hamiltonian satisfying Assumption (DG). For any cut and any $\delta = \text{poly}^{-1}(n)$ there is subspace $S \subseteq \mathcal{H}$*

that is δ-close to T and such that every element of S has Schmidt Rank no larger than

$$s(\delta) = \tilde{O}(r^2)e^{\tilde{O}\left(\frac{1}{\gamma}\log^3 d\right)} \cdot e^{\tilde{O}\left(\gamma^{-1/4}\log^{3/4}(\frac{1}{\delta})\log d\right)}.$$

Moreover, every state $|\psi\rangle \in T$ has entanglement entropy

$$S(|\psi\rangle\langle\psi|) \leq 4\log r + \tilde{O}\Big(\frac{1}{\gamma}\log^3 d\Big)$$

and can be approximated by a state $|\psi'\rangle$ such that $|\langle\psi|\psi'\rangle| > 1-\delta$ and $|\psi'\rangle$ has an MPS representation with bond dimension bounded by

$$\tilde{O}(r^2)e^{\tilde{O}\left(\frac{1}{\gamma}\log^3 d\right)}e^{\tilde{O}\left(\gamma^{-1/4}\log^{3/4}(\frac{n}{\delta})\log d\right)}.$$

We note that the dependence on r in the bounds for the Schmidt rank and the bond dimension of the MPS approximation can be improved from $\tilde{O}(r^2)$ to $r^{1+o(1)}$. In fact there is a simpler way of getting these bounds through a clean "bootstrapping" argument, for which we refer to the full version [2].

4 Moving to algorithms

There are two main obstacles to turning Procedure 1 into an efficient algorithm. The first consists in showing that operators $\{A_i\}$ satisfying the conditions of Theorem 9 can be generated efficiently from a description of the Hamiltonian, and that it is possible to apply these operators efficiently, as required to complete the error reduction step of the **Merge** process. The following theorem states that this can be achieved.

▶ **Theorem 12** (Efficient AGSP, (DG)). *There exists a procedure* **Generate**(H, M, ε'_M) *which takes as input*

- *A local Hamiltonian H satisfying Assumption (DG),*
- *A decomposition $\mathcal{H} = \mathcal{H}_L \otimes \mathcal{H}_M \otimes \mathcal{H}_R$ of the n-qudit space into contiguous blocks,*
- *An estimate ε'_M for the minimal energy ε_M of the restriction of H to $\mathcal{H}_M$ such that $|\varepsilon_M - \varepsilon'_M| \leq 10$,*

and returns

- *MPO representations for a collection of D^2 operators $\{A_i\}_{i=1}^{D^2}$ acting on $\mathcal{H}_M$ and of bond dimension at most $n^{\tilde{O}(\gamma^{-2})}$ satisfying the conditions of Theorem 9 for some subspace $\tilde{T}$,*
- *An MPO for an operator $\tilde{H}_M$ such that $\|\tilde{H}_M\| = O(\gamma^{-1}\log\gamma^{-1})$ and the minimal energy $\tilde{\varepsilon}_M$ of $\tilde{H}_M$ restricted to $\tilde{T}$ satisfies $|\varepsilon_M - \tilde{\varepsilon}_M| < 1/2$.*

Moreover, **Generate**(H, M, ε'_M) *runs in time $n^{\tilde{O}(\gamma^{-2})}$.*[2]

The proof of Theorem 12 relies on new constructions of approximate ground state projections (AGSP), details of which are given in the full version [2]. The theorem guarantees that the $\{A_i\}$ can be constructed efficiently *provided* it is possible to provide a good approximation to the ground state energy of the restriction of H to $\mathcal{H}_M$. This is the reason for including $\tilde{H}_M$ as parts of the output of **Generate** , which is then used by an additional step of *energy estimation* incorporated in Algorithm 2.

[2] Here and in all our estimates on running times we suppress dependence on the local dimension d, which is treated as a constant.

Algorithm 2: Given a local Hamiltonian H satisfying Assumption (GS), returns a set $V_1^{\log n}$ that is 0.015-close to the ground space T of H.

Initialization: Set $V_j^0 = \mathcal{H}_j$ and $\varepsilon'_{0,j} = 0$ for $j \in \{1, 2, \dots n\}$.

Iteration:

For $i = 1, \dots, \log(n)$ and $j \in \{1 \dots \frac{n}{2^i}\}$ do:

Generate. Let $M = \{(j-1)2^i, (j-1)2^i + 1, \dots, j2^i - 1\}$, $\varepsilon'_M = \varepsilon'_{i-1,2j-1} + \varepsilon'_{i-1,2j}$. Set $(\{A_i\}, \tilde{H}_M) = \mathbf{Generate}(H, M, \varepsilon'_M)$.

Merge. Set $V_j^i = \mathbf{Merge'}(V_{2j-1}^{i-1}, V_{2j}^{i-1}, \{A_i\}, s, k, \xi) \subseteq \mathcal{H}_{[(j-1)2^i+1, j2^i]}$, where s and k are specified in Theorem 10 and $\xi = \text{poly}^{-1}(n, r)$ is chosen small enough (see proof of Theorem 13).

New Energy Estimation. Form the subspace $V = \{A_i\}^t \cdot (V_{2j-1}^{i-1} \otimes V_{2j}^{i-1})$ for $t = \Theta(\log \gamma^{-1})$. Compute the smallest eigenvalue $\varepsilon'_{i,j}$ of the restriction of $\tilde{H}_M$ to V.

Final step: Return $V_1^{\log n}$.

The second difficulty encountered in turning Procedure 1 into an algorithm is that, even if the $\{A_i\}$ can be applied efficiently, due to the logarithmic number of iterations it may be that the bond dimension of MPS representations for the elements of the viable sets we work with increase to super-polynomial. This difficulty can be overcome by introducing a *bond trimming* component Trim_ξ to the **Merge** procedure, resulting in the following modified procedure **Merge**' taking an additional trimming parameter ξ as input (ξ will usually be of order $\text{poly}^{-1}(n)$):

Merge'$(V_1, V_2, \{A_i\}, s, k, \xi)$:

Step 1: Tensoring. Set $W = V_1 \otimes V_2$.

Step 2: Random Sampling. Let $W' \subseteq W$ be a random s-dimensional subspace of W.

Step 3: Error Reduction. Set $V = W'$. Repeat k times:

Set $V = \text{Trim}_\xi(\text{Span}(\cup_i A_i V))$.

Return V.

Correctness of **Merge**' (for an appropriate choice of ξ) is based on the area law proven in Corollary 11. The details of the trimming[3] procedure Trim_ξ, together with the analysis of **Merge**', are given in the full paper [2].

The following theorem proves the correctness of Algorithm 2.

▶ **Theorem 13.** *Let H be a local Hamiltonian satisfying Assumption (DG). Then with probability at least $1 - \frac{1}{n}$ the set $V_1^{\log n}$ returned by Algorithm 2 is 0.015-viable for T.*[4] *The running time of the algorithm is $n^{\tilde{O}(\gamma^{-2})}$.*

Proof. The proof mirrors the analysis of Procedure 1 given in the proof of Theorem 5 in Section 3.3; the two main differences are that we must show that at every step, with high enough probability the call to **Merge**' yields a good viable set and the **New Energy Estimation** step yields a sufficiently accurate energy estimate for the next iteration.

[3] We note that the trimming procedure differs from that of [8]

[4] The probability of success can be improved to $1 - \text{poly}^{-1}(n)$ by scaling the parameter s used in the algorithm by an appropriate constant.

Both conditions are satisfied at initialization since each V_j^0 is 0-viable for T with $|V_j^0| = d$ and the energy estimates are accurate since there are no terms of the Hamiltonian when restricting to single particles.

Assume V_{2j-1}^{i-1} and V_{2j}^{i-1} are both .015-viable for T with $|V_{2j-1}^{i-1}|, |V_{2j}^{i-1}| \leq q$, and $\varepsilon'_{i-1,2j-1}$, $\varepsilon'_{i-1,2j}$ both within an additive ± 3 of their respective true values (the ground state energy of the restriction of H to the corresponding spaces). As a result ε'_M is within 7 of the correct value ε_M, and by Theorem 12 **Generate** yields a set $\{A_i\}$ with the properties stated in Theorem 9. Thus by Theorem 10 V_j^i is .015-viable for T with probability $1 - e^{-\Omega(s)} \geq 1 - \frac{1}{n^2}$ (provided $r \geq \log n$, which we may always assume without loss of generality). For this we need to check Theorem 10 still applies when **Merge** is replaced by **Merge**' . The analysis of the trimming procedure given in [2] shows that this is the case provided the error reduction parameter Δ associated with the $\{A_i\}$ is replaced by $(\Delta + 2\sqrt{krs}\xi)$; choosing $\xi = \text{poly}^{-1}(n, r)$ we may ensure that $2\sqrt{krs}\xi < .0001D^{-12}$. With this choice, the remaining calculation of 10 applies to still yields that V_j^i is .015-viable for T.

Once this has been established, an application of the third item from Theorem 9 shows that provided the constant implicit in the definition of t is chosen large enough the subpace V obtained after the **New Energy Estimation** step is $O(\gamma^2)$-viable for $\tilde{T}$. Using that $\|\tilde{H}_M\| = O(\gamma^{-1} \log \gamma^{-1})$ it follows that $\varepsilon'_{i,j}$ is within an arbitrarily small constant of the minimal energy of $\tilde{H}_M$ restricted to $\tilde{T}$. Using the last guarantee from Theorem 12, $\varepsilon'_{i,j}$ is within $\frac{3}{2}$ of the minimal energy ε_M of the restriction of H to $\mathcal{H}_M$. This completes the inductive step.

We have shown that the iterative step succeeds with probability at least $1 - 1/n^2$; since there are a total of n such merging steps, applying a union bound the final set $V_1^{\log n}$ is .015-viable with probability at least $1 - \frac{1}{n}$.

In total the complete algorithm requires only a polynomial number of operations on MPS representations of vectors. Due to trimming, all these vectors have polynomial bond dimension and thus each operation can be implemented in polynomial time. The complexity is dominated by the complexity of the procedure **Generate** and the application of the operators A_i, which is $n^{\tilde{O}(\gamma^{-2})}$. ◀

We end this section by noting that in case one desires a better than constant approximation to T the final step of Algorithm 2 can be replaced by the following:

Final step. Set $K = (\mathbb{1} - H/\|H\|)$ and $\tau = 10\|H\|\gamma^{-1} \log(1/\delta)$. Choose an orthonormal basis $\{|y_i^{(0)}\rangle\}$ for $V_1^{\log n}$. Repeat for $t = 1, \ldots, \tau$:

Set $\{|y_i^{(t)}\rangle\} = \text{Trim}_\xi(\text{Span}\{K|y_i^{(t-1)}\rangle\})$.

Return $\{|z_i\rangle\}$, the smallest r eigenvectors of H restricted to $W = \text{Span}\{|y_i^{(\tau)}\rangle\}$.

The result of this step is a basis $\{|z_i\rangle\}$ for a subspace S such that $\angle_m(S, T) \leq \delta$.

Frustration-free Hamiltonians with a unique ground state

The computation-intensive step of the AGSP-based RG transformation introduced in Section 3 is the construction and subsequent application of the set of operators $\{A_i\}$. In the special case where the Hamiltonian H satisfies Assumption (FF), i.e. H is frustration-free and has a spectral gap, the operators $\{A_i\}$ can be constructed very efficiently, yielding strong bounds on the running time. We state a specialized theorem for this setting. The proof is given in the full version [2].

Algorithm 3: Given a local Hamiltonian H satisfying Assumption (FF), returns a δ-approximation to its ground state $|\Gamma\rangle$.

Initialization: Set $V_j^0 = \mathcal{H}_j$ for $j \in \{1, 2, \dots n\}$.

Iteration:

For $i = 1, \dots, \log(n)$ and all $j \in \{1, \dots, \frac{n}{2^i}\}$,

Generate. Let $M = \{(j-1)2^i, (j-1)2^i + 1, \dots, j2^i - 1\}$.

Set $\{A_i\} =$ **Generate 2**(H, M).

Merge. Set $V_j^i =$ **Merge'**$(V_{2j-1}^{i-1}, V_{2j}^{i-1}, \{A_i\}, s, k, \xi) \subseteq \mathcal{H}_{[(j-1)2^i+1, j2^i]}$, where k, s are as in Theorem 10 (with $r = 1$) and $\xi = \tilde{\Theta}(n^{-1/2})$.

Final step:

Let K be the unique operator A computed at the last iteration, and $\tau = 10\|H\|\gamma^{-1}\log(1/\delta)$. Choose an orthonormal basis $\{|y_i^{(0)}\rangle\}$ for $V_1^{\log n}$. Repeat for $t = 1, \dots, \tau$:

Set $\{|y_i^{(t)}\rangle\} = \mathrm{Trim}_\xi(\mathrm{Span}\{|Ky_i^{(t-1)}\rangle\})$, for $\xi = \tilde{\Theta}(n^{-1/2})$.

Return the smallest eigenvector $|z\rangle$ of H restricted to $W = \mathrm{Span}\{|y_i^{(\tau)}\rangle\}$.

▶ **Theorem 14** (Efficient AGSP, (FF))**.** *Let H be a local Hamiltonian satisfying Assumption (FF), and $\mathcal{H} = \mathcal{H}_L \otimes \mathcal{H}_M \otimes \mathcal{H}_R$ a decomposition of the n-qudit space into contiguous regions. There exists a procedure* **Generate 2**(H, M) *which takes as input*

- *A local Hamiltonian H satisfying Assumption (DG),*
- *A decomposition $\mathcal{H} = \mathcal{H}_L \otimes \mathcal{H}_M \otimes \mathcal{H}_R$ of the n-qudit space into contiguous blocks,*

and returns MPO representations for a collection of D^2 operators $\{A_i\}_{i=1}^{D^2}$ acting on $\mathcal{H}_M$ such that the following hold:

- $D = 2^{\tilde{O}(\gamma^{-1}\log^3 d)}$
- *There is $\Delta > 0$ such that $D^{12}\Delta < \frac{1}{2000}$ and for any $S \subseteq \mathcal{H}_M$ that is δ-viable for $\{|\Gamma\rangle\}$ it holds that $S' = \mathrm{Span}\{\cup_i A_i S\}$ is δ'-viable for T with $\delta' = \frac{\Delta}{(1-\delta)^2}$.*
- *Each A_i has bond dimension at most $2^{\tilde{O}(\gamma^{-2}\log^5 d)}$.*

Moreover, for constant d and $\gamma > 0$ the procedure **Generate 2**(H, M, ε'_M) *runs in time $n^{(1+o(1))}$.*

We note that in the case where M consists of all n qudits the procedure returns a single operator A acting on the whole space. Algorithm 3 is an adaptation of Algorithm 2 to the case of frustration-free Hamiltonians.

▶ **Theorem 15.** *Let H be a local Hamiltonian satisfying Assumption (FF) and $\delta = n^{-\omega(1)}$. With probability at least $1 - \frac{1}{n}$ the vector $|z\rangle$ returned by Algorithm 3 is such that $|\langle z|\Gamma\rangle| \geq 1 - \delta$. Moreover the algorithm runs in time $O(n^{1+o(1)}\mathrm{M}(n))$, where $\mathrm{M}(n) = O(n^{2.38})$ denotes matrix multiplication time.*

The proof follows the same outline as that of Theorem 13 analyzing Algorithm 2, and we refer to the full version [2] for details. We note that in [7], Huang gives a very simple, though less efficient, algorithm for this frustration free case, by adding one particle at a time and within each step randomly projecting onto a one dimensional space followed by application of an AGSP.

Acknowledgements. We thank Andras Molnar for comments on an earlier draft of this paper, and Christopher T. Chubb for comments and the permission to include the suggestive pictures representing the tensor network structure of the isometry produced by our algorithms.

References

1. Dorit Aharonov, Daniel Gottesman, Sandy Irani, and Julia Kempe. The power of quantum systems on a line. *Communications in Mathematical Physics*, 287(1):41–65, 2009. doi:10.1007/s00220-008-0710-3.
2. Itai Arad, Zeph Landau, Umesh Vazirani, and Thomas Vidick. Rigorous RG algorithms and area laws for low energy eigenstates in 1D. *arXiv preprint arXiv:1602.08828*, 2016. URL: https://arxiv.org/abs/1602.08828.
3. Christopher T. Chubb and Steven T. Flammia. Computing the Degenerate Ground Space of Gapped Spin Chains in Polynomial Time. *ArXiv e-prints*, February 2015. arXiv:1502.06967.
4. J. Eisert, M. Cramer, and M. B. Plenio. Colloquium: Area laws for the entanglement entropy. *Rev. Mod. Phys.*, 82(1):277–306, Feb 2010. doi:10.1103/RevModPhys.82.277.
5. Matthew B. Hastings. An area law for one-dimensional quantum systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(08):P08024, 2007. URL: http://stacks.iop.org/1742-5468/2007/i=08/a=P08024.
6. Yichen Huang. A polynomial-time algorithm for the ground state of one-dimensional gapped Hamiltonians. *ArXiv e-prints*, June 2014. arXiv:1406.6355.
7. Yichen Huang. A simple efficient algorithm in frustration-free one-dimensional gapped systems. *ArXiv e-prints*, October 2015. arXiv:1510.01303.
8. Zeph Landau, Umesh Vazirani, and Thomas Vidick. A polynomial time algorithm for the ground state of one-dimensional gapped local hamiltonians. *Nature Physics*, 2015. arXiv:1307.5143.
9. Lluís Masanes. Area law for the entropy of low-energy states. *Physical Review A*, 80(5):052104, 2009.
10. Pankaj Mehta and David Schwab. An exact mapping between the variational renormalization group and deep learning. Technical report, arXiv preprint arXiv:1410.3831, 2014. arXiv:1410.3831.
11. Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010. arXiv:1011.3027.
12. G. Vidal. Class of quantum many-body states that can be efficiently simulated. *Phys. Rev. Lett.*, 101:110501, Sep 2008. doi:10.1103/PhysRevLett.101.110501.
13. Guifre Vidal. Entanglement renormalization: an introduction. *arXiv preprint arXiv:0912.1651*, 2009.
14. Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863–2866, Nov 1992. doi:10.1103/PhysRevLett.69.2863.
15. Kenneth G Wilson. The renormalization group: Critical phenomena and the kondo problem. *Reviews of Modern Physics*, 47(4):773, 1975.

The Flow of Information in Interactive Quantum Protocols: the Cost of Forgetting*†

Mathieu Laurière[‡1] **and Dave Touchette**[§2]

1 **NYU-ECNU Institute of Mathematical Sciences at NYU Shanghai, China**
mathieu.lauriere@gmail.com

2 **Institute for Quantum Computing and Department of Combinatorics and Optimization, University of Waterloo, and Perimeter Institute for Theoretical Physics, Waterloo, Canada**
touchette.dave@gmail.com

Abstract

In two-party interactive quantum communication protocols, we study a recently defined notion of quantum information cost (QIC), which has most of the important properties of its classical analogue (IC). Notably, its link with amortized quantum communication complexity has been used to prove an (almost) tight lower bound on the bounded round quantum complexity of Disjointness. However, QIC was defined through a purification of the input state. This is valid for fully quantum inputs and tasks but difficult to interpret even for classical tasks. Also, its link with other notions of information cost that had appeared in the literature was not clear.

We settle both these issues: for quantum communication with classical inputs, we characterize QIC in terms of information about the input registers, avoiding any reference to the notion of a purification of the classical input state. We provide an operational interpretation of this new characterization as the sum of the costs of revealing and of forgetting information about the inputs. To obtain this result, we prove a general Information Flow Lemma assessing the transfer of information in general interactive quantum processes. Specializing this lemma to interactive quantum protocols accomplishing classical tasks, we are able to demistify the link between QIC and other previous notions of information cost in quantum protocols. Furthermore, we clarify the link between QIC and IC by simulating quantumly classical protocols.

Finally, we apply these concepts to argue that any quantum protocol that does not forget information solves Disjointness on n-bits in $\Omega(n)$ communication, completely losing the quadratic quantum speedup. Hence forgetting information is here a necessary feature in order to obtain any significant improvement over classical protocols. We also prove that QIC at 0-error is exactly n for Inner Product, and $n(1 - o(1))$ for a random Boolean function on $n + n$ bits.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes, E.4 Coding and Information Theory

Keywords and phrases Communication Complexity, Information Complexity, Quantum Computation and Information

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.47

* A long version of this work can be found on arXiv.org, see https://arxiv.org/abs/1701.02062.

† The authors are very grateful to Anurag Anshu, André Chailloux, Ankit Garg, Iordanis Kerenidis, Ashwin Nayak, and Penghui Yao for many useful discussions.

‡ M.L. has been supported by ERC grant QCC. D.T. is supported in part by NSERC, CIFAR, Industry Canada and ARL CDQI program. IQC and PI are supported in part by the Government of Canada and the Province of Ontario. Part of this research was conducted while M.L. was a PhD student with the Institut de Recherche en Informatique Fondamentale, Université Paris Diderot

§ Part of this research was conducted while D.T. was a PhD student with the Département d'informatique et de recherche opérationnelle, Université de Montréal and was supported in part by a FRQNT B2 Doctoral research scholarship, and by CryptoWorks21.

© Mathieu Laurière and Dave Touchette;
licensed under Creative Commons License CC-BY

Overlapping Qubits*

Rui Chao[1], Ben W. Reichardt[2], Chris Sutherland[3], and Thomas Vidick[4]

1 **University of Southern California, Los Angeles, USA**
ruichao@usc.edu
2 **University of Southern California, Los Angeles, USA**
ben.reichardt@usc.edu
3 **University of Southern California, Los Angeles, USA**
cjsuther@usc.edu
4 **Caltech, Pasadena, USA**
vidick@cms.caltech.edu

Abstract

An ideal system of n qubits has 2^n dimensions. This exponential grants power, but also hinders characterizing the system's state and dynamics. We study a new problem: the qubits in a physical system might not be independent. They can "overlap," in the sense that an operation on one qubit slightly affects the others.

We show that allowing for slight overlaps, n qubits can fit in just polynomially many dimensions. (Defined in a natural way, all pairwise overlaps can be $\leq \epsilon$ in $n^{O(1/\epsilon^2)}$ dimensions.) Thus, even before considering issues like noise, a real system of n qubits might inherently lack any potential for exponential power.

On the other hand, we also provide an efficient test to certify exponential dimensionality. Unfortunately, the test is sensitive to noise. It is important to devise more robust tests on the arrangements of qubits in quantum devices.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases Quantum computing, Qubits, Dimension test

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.48

1 Introduction

Quantum computers start with the qubit, a two-level quantum system. They achieve their power by combining many qubits. A system of n independent qubits is associated to a 2^n-dimensional tensor-product space, $(\mathbf{C}^2)^{\otimes n}$, and quantum algorithms exploit this exponential dimensionality. However, with great power also comes great guile. In experiments, it is exceedingly difficult to characterize the states and dynamics of large quantum systems. An efficient test, running in polynomial time, can only probe a limited portion of an exponentially complex system.

Before getting to state or process tomography, however, there is the problem of characterizing the system's Hilbert space, and the arrangement of the qubits within it. In particular, what if the qubits are not in tensor product, but "overlap," so an operation on one qubit

* R.C., B.R. and C.S. supported by NSF grant CCF-1254119 and ARO grant W911NF-12-1-0541. T.V. supported by NSF CAREER grant CCF-1553477, an AFOSR YIP award, and the IQIM, an NSF Physics Frontiers Center (NFS Grant PHY-1125565) with support of the Gordon and Betty Moore Foundation (GBMF-12500028).

© Rui Chao, Ben W. Reichardt, Chris Sutherland, and Thomas Vidick;
licensed under Creative Commons License CC-BY

can slightly affect the others? Given a system that supposedly has n independent qubits, how can we efficiently test that there really are 2^n dimensions? Unfortunately, we show that very small systems, with only polynomially many dimensions, can contain n qubits that are nearly pairwise independent, i.e., an operation on qubit i can have only a small effect on qubit j for all $i \neq j$. In fact, there are particular states in n^2-dimensional systems for which n qubits look to be exactly pairwise independent, in tensor product. (We will give more technical statements of these results in a moment.)

The issue of overlapping qubits is a new concern for the characterization of quantum devices. A common complaint about today's quantum devices, especially those targeted at adiabatic quantum optimization or quantum annealing, is that it is difficult even to verify their quantum-ness [1]. High noise rates can decohere systems, making them classical. Our examples raise a different problem: a system might indeed be quantum mechanical and even look like it has many qubits, but still quantum power is lacking because the system is low-dimensional.

On the other hand, we show that low-dimensional systems cannot totally fool us. First, if all pairs among n qubits are sufficiently close to being independent, then in fact there are nearby qubits that are exactly independent (in tensor product); and hence the dimension must be at least 2^n. Second, we provide a test for independence, one that efficiently checks not just pairwise interactions but n-wise interactions, and thereby can verify that the system dimension is almost 2^n. The test only involves measuring the qubits one at a time, so it is conceivably practical—except it is still sensitive to noise.

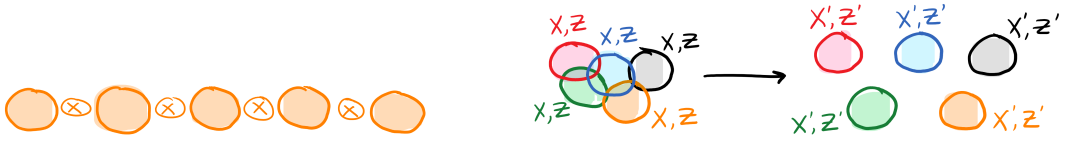
Overlapping qubits

The concept of overlapping, dependent qubits is not standard in quantum information theory. In general, multiple qubits are always assumed to be in tensor product; in common usage n qubits directly means $(\mathbf{C}^2)^{\otimes n}$. However, though it may be invisibly built into our notation and habits of thought, this is in fact an independence assumption, which needs to be justified. Precisely, then, what is a qubit, and what does it mean for two qubits to overlap?

1. What is a qubit? A qubit in a space $\mathcal{H}$ is a two-dimensional register in tensor product with the rest of the space. That is, from an isomorphism between $\mathcal{H}$ and $\mathbf{C}^2 \otimes \mathcal{H}'$, the $\mathbf{C}^2$ register defines a qubit. Since the basis for $\mathcal{H}'$ does not matter, instead of specifying the isomorphism it is more convenient to work in the dual Heisenberg picture, in which a qubit is defined through the observables that act on it, an algebra generated by the four Pauli matrices. In fact, a pair of norm-one observables X and Z that anti-commute suffice to define a qubit; it is then possible to choose a basis in which $X = \sigma^x \otimes \mathbf{1}_{\mathcal{H}'}$ and $Z = \sigma^z \otimes \mathbf{1}_{\mathcal{H}'}$, where σ^x and σ^z are the standard Pauli operators (see Lemma 2.2).
2. Two qubits are independent, or in tensor product, when all operators on the qubits commute. Thus n qubits, defined by anti-commuting X_j, Z_j for $j = 1, \ldots, n$, are pairwise independent if $[X_i, X_j] = [X_i, Z_j] = [Z_i, Z_j] = 0$ for all $i \neq j$. It follows that there is a change of basis under which $\mathcal{H} = (\mathbf{C}^2)^{\otimes n} \otimes \mathcal{H}'$ and $X_j = \sigma_j^x \otimes \mathbf{1}_{\mathcal{H}'}$, $Z_j = \sigma_j^z \otimes \mathbf{1}_{\mathcal{H}'}$ (Theorem 2.3).

When are two qubits "almost" independent? For qubits specified by reflections X_1, Z_1 and X_2, Z_2, how close they are to lying in tensor product can be measured by the largest commutator norm, $\max_{S,T \in \{X,Z\}} \|[S_1, T_2]\|$.

Almost independence is a useful concept because in reality one can never probe for the existence of n independent qubits. The exact tensor-product structure of a Hilbert space cannot be experimentally tested. Due to inevitable measurement imprecision, one



(a) (b)

Figure 1 (a) A qubit is a two-dimensional system in tensor product with the rest of the space. Qubits "overlap" if the corresponding Pauli operators do not commute. When their Pauli operators do commute, the qubits are in tensor product with each other (Theorem 2.3). (b) We ask how many qubits can be packed into a 2^n dimensional space with small pairwise overlap. For a lower bound, we give a randomized construction, based on the Johnson-Lindenstrauss Lemma and fermion algebra (Theorem 3.1). For an upper bound, we separate qubits with small pairwise overlap, finding nearby qubits with zero overlap (Theorem 3.6).

could at best hope to show approximate relations, like $\|[S_i, T_j]\| \leq \epsilon$. This concept is also mathematically well-motivated. It amounts to studying approximate representations of the n-qubit Pauli group.[1] It can alternatively be tied to questions on the stability of relations defining the Pauli algebra [10].

Our results

We begin by asking: how many overlapping qubits can be packed into 2^n dimensions? We prove both lower and upper bounds. Of course, only n independent qubits fit.

For the lower bound, we give a randomized construction, based on the Johnson-Lindenstrauss lemma, for packing many nearly orthogonal unit vectors, and on the exterior algebra. We show that exponential in n many qubits can be packed with pairwise overlaps $\|[S_i, T_j]\|$ of order $\sqrt{(\log n)/n}$. In general, for overlaps $\|[S_i, T_j]\| \leq \epsilon$, $e^{O(n\epsilon^2)}$ qubits can be packed into 2^n dimensions; see Theorem 3.1. Parameterized differently, the construction places n ϵ-overlapping qubits in only $n^{O(1/\epsilon^2)}$ dimensions.

Note that this construction does not allow for compressing information. Even though exponentially many nearly independent qubits can be packed into $(\mathbf{C}^2)^{\otimes n}$, this does not allow for reliably storing more than n bits, and thus does not violate Nayak's private information retrieval bound [14]. If one tried to store $\gg n$ bits into $(\mathbf{C}^2)^{\otimes n}$ by putting a bit into each of the embedded qubits, one at a time, by the end the early bits would be unrecoverable because of accumulated errors.

For the upper bound, we show that even allowing pairwise overlaps $\|[S_i, T_j]\|$ as large as c/n, for a certain constant c, there is still room only for n qubits in 2^n dimensions. The precise statement is in Theorem 3.6. The proof constructively extracts n independent qubits from n overlapping qubits. The key difficulty is to ensure that errors do not explode; naively separating, say, the second qubit from the first could double its overlap with each of the remaining qubits, yielding an unmanageable exponential blow-up in the total displacement needed to separate the qubits. See Figure 1.

The construction in the upper bound loses a factor of n, and we give an example to show that this is necessary (Lemma 3.9). Yet there is still a gap between our lower and upper bounds. For the range of overlaps $1/n \lesssim \epsilon \lesssim \sqrt{(\log n)/n}$, we do not know whether strictly more than n qubits can be packed into 2^n dimensions.

[1] We caution that there does not seem to be a standard definition for an approximate group representation in the mathematical literature; see, e.g., [3, 13] for work in this direction.

Given access to an experimental system, it is difficult to imagine tests for determining $\|[S_i, T_j]\|$. The problem is that the quantum system can be in an unknown state $|\psi\rangle$, and we can only learn about operators' effects on $|\psi\rangle$. If S_i and T_j are far from commuting, but only on a portion of the Hilbert space in which $|\psi\rangle$ has no support, this is undetectable. In Section 4, we therefore consider a *state-dependent* overlap measure. This is the same measure that is used in results on self-testing such as [11, 12], and it is the relevant measure for applications to device-independent cryptography [8]. Note however that our setting differs from the usual one in self-testing, as we do not assume any a priori bipartite structure on the Hilbert space.

We first give a practical protocol for testing if $\|[S_i, T_j]|\psi\rangle\| \approx 0$: measure S_i, measure T_j, then measure S_i again and check that it gives the same result. However, this test is not enough; we give a construction of a state and n qubit operators in $< n^2$ dimensions, such that for $i \neq j$, $[S_i, T_j]|\psi\rangle = 0$ exactly. Finally we give a more advanced test that efficiently checks not just pairwise commutation relationships, like $[S_i, T_j]|\psi\rangle \approx 0$, but also higher-order relationships like $S_i T_j U_k|\psi\rangle \approx U_k T_j S_i|\psi\rangle$. This test can verify that the system dimension is almost 2^n.

2 What is a qubit? When are qubits in tensor product?

As explained in the introduction, we take a basis-independent, operator-centric view of what it means to have a qubit, or multiple independent qubits, in an a priori unstructured Hilbert space $\mathcal{H}$. The following definition formalizes these notions. Notation: Let $[n] = \{1, 2, \ldots, n\}$, and $I = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$, $\sigma^x = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $\sigma^y = \left(\begin{smallmatrix} 0 & -i \\ i & 0 \end{smallmatrix}\right)$ and $\sigma^z = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$ be the Pauli matrices. The commutator is $[S, T] = ST - TS$, and the anticommutator is $\{S, T\} = ST + TS$. When we write, e.g., "S_j for $S \in \{X, Z\}$" we mean the set $\{X_j, Z_j\}$, i.e., the letter S is meant to be directly replaced by X or Z.

▶ **Definition 2.1.** A *qubit* in a Hilbert space $\mathcal{H}$ is a pair of anti-commuting reflections (X, Z) on $\mathcal{H}$. The *overlap* between two qubits (X_1, Z_1) and (X_2, Z_2) is given by $\max_{S,T \in \{X,Z\}} \|[S_1, T_2]\|$. The qubits are in *tensor product* if they have overlap 0; in this case we also say that the qubits are *independent*.

The following simple lemma ties this definition to the more usual one of a qubit as defined by a factorization $\mathcal{H} \simeq \mathbf{C}^2 \otimes \mathcal{H}'$. The lemma is a special case of Theorem 2.3 below.

▶ **Lemma 2.2.** *Let X and Z be reflections (Hermitian operators that square to the identity) on a separable Hilbert space $\mathcal{H}$ such that X and Z anti-commute: $\{X, Z\} = 0$. Then there exists a separable space $\mathcal{H}'$ such that $\mathcal{H}$ is isomorphic to $\mathbf{C}^2 \otimes \mathcal{H}'$, and up to a unitary change of basis the reflections X, Z are the standard Pauli operators:*

$$X = \sigma^x \otimes \mathbf{1}_{\mathcal{H}'}, \qquad Z = \sigma^z \otimes \mathbf{1}_{\mathcal{H}'} \ .$$

The following theorem justifies our definition of two qubits being in "tensor product" when their overlap is 0, or equivalently when the associated reflections pairwise commute.

▶ **Theorem 2.3.** *Suppose that $X_1, Z_1, \ldots, X_n, Z_n$ are reflections on $\mathcal{H}$ such that for all j, $\{X_j, Z_j\} = 0$ and furthermore for all $i \neq j$ and $S, T \in \{X, Z\}$, S_i and T_j pairwise commute, $[S_i, T_j] = 0$. Then there exists a separable space $\mathcal{H}''$ such that $\mathcal{H}$ is isomorphic to $(\mathbf{C}^2)^{\otimes n} \otimes \mathcal{H}''$, and up to a unitary change of basis the reflections X_j, Z_j are the standard Pauli operators on n qubits:*

$$X_1 = \sigma^x \otimes I^{\otimes(n-1)} \otimes \mathbf{1}_{\mathcal{H}''} \qquad \cdots \qquad X_n = I^{\otimes(n-1)} \otimes \sigma^x \otimes \mathbf{1}_{\mathcal{H}''}$$
$$Z_1 = \sigma^z \otimes I^{\otimes(n-1)} \otimes \mathbf{1}_{\mathcal{H}''} \qquad \cdots \qquad Z_n = I^{\otimes(n-1)} \otimes \sigma^z \otimes \mathbf{1}_{\mathcal{H}''} \ .$$

Proof. Let $X = X_1$, $Z = Z_1$. As $Z^2 = \mathbf{1}$, $\Pi_\pm = \frac{1}{2}(\mathbf{1} \pm Z)$ are projections, with $\Pi_+ + \Pi_- = \mathbf{1}$, $\Pi_+ - \Pi_- = Z$ and $\Pi_+\Pi_- = \Pi_-\Pi_+ = 0$. Multiplying both sides of $\{X, Z\} = 0$ by $\Pi_\pm$ yields $\Pi_\pm X \Pi_\pm = 0$, i.e., $X = \Pi_+ X \Pi_- + \Pi_- X \Pi_+$. Then $X^2 = \mathbf{1}$ implies that $\Pi_\pm X \Pi_\mp X \Pi_\pm = \Pi_\pm$; and comparing the ranks of both sides gives $\mathrm{Rank}(\Pi_\mp) \geq \mathrm{Rank}(\Pi_\pm)$, i.e., $\mathrm{Rank}(\Pi_+) = \mathrm{Rank}(\Pi_-)$.

Let $|u_1^\pm\rangle, |u_2^\pm\rangle, \ldots$ be an orthonormal basis for $\mathrm{Range}(\Pi_\pm)$. Let $S = \sum_j (|u_j^+\rangle\langle u_j^-| + |u_j^-\rangle\langle u_j^+|)$. Then $S = S^\dagger$, $S^2 = \mathbf{1}$ and $S\Pi_\pm = \Pi_\mp S$. Let $U = \Pi_+ X \Pi_- S + \Pi_-$. U is unitary: $UU^\dagger = U^\dagger U = \mathbf{1}$. Furthermore, $U^\dagger Z U = Z$, and $U^\dagger X U = S$. Relabeling the basis elements $|0, j\rangle = |u_j^+\rangle$, $|1, j\rangle = |u_j^-\rangle$, we obtain $U^\dagger Z U = \sigma^z \otimes \mathbf{1}$ and $U^\dagger X U = \sigma^x \otimes \mathbf{1}$, as desired.

Now consider X_2. In the above basis, it can be expanded as $I \otimes A + \sum_{\beta \in \{x,y,z\}} \sigma^\beta \otimes B_\beta$, but the commutation relationships $[X_2, X_1] = [X_2, Z_1] = 0$ imply that each $B_\beta = 0$. Similarly, all the reflections $Z_2, \ldots, X_n, Z_n$ act trivially on the first $\mathbf{C}^2$ register. Inductively repeating the above argument for X_1 and Z_1 gives the theorem. ◀

Registers that are in tensor product are independent of each other, in the sense that for a quantum state $|\psi\rangle \in \mathcal{H}' \otimes \mathcal{H}''$, a quantum operation on $\mathcal{H}'$ cannot affect the reduced density matrix $\mathrm{Tr}_{\mathcal{H}'} |\psi\rangle\langle\psi|$ in the other register. It should be noted, though, that a qubit can simultaneously have maximal overlap with many other mutually independent qubits. For example, for n odd, $X = (\sigma^x)^{\otimes n}$ and $Z = (\sigma^z)^{\otimes n}$ are anti-commuting reflections, defining a qubit, such that for every $j \in [n]$, $\|[X, \sigma_j^z]\| = \|[Z, \sigma_j^x]\| = 2$. (Similarly, in $(\mathbf{C}^2)^{\otimes n}$, for a Haar random unitary U, $\|[U\sigma_1^\alpha U^\dagger, \sigma_j^\beta]\|$ will be concentrated around the maximal value of 2.) Thus the norm of the reflections' commutator is not a "monogamous" measure of qubit overlap.

3 Packing qubits

How many pairwise ϵ-overlapping qubits can be packed into 2^n dimensions? Formally, in 2^n dimensions, we wish to place $2m$ reflections $(X_1, Z_1), \ldots, (X_m, Z_m)$ such that each pair (X_j, Z_j) defines a qubit, so that $\{X_j, Z_j\} = 0$, and operators with different indices nearly commute: $\|[S_i, T_j]\| \leq \epsilon$ for $i \neq j$ and $S, T \in \{X, Z\}$. How large can m be?

One's intuition might be pulled in either of two directions. From the perspective of information theory, Nayak's private information retrieval bound $m \leq n/(1 - H(p))$ [14] suggests that packing $\omega(n)$ qubits into 2^n dimensions is unlikely to be possible. However, a formal connection between our problem and private information retrieval is not obvious: the existence of m pairs of approximately commuting qubit operators does not imply that there exists a family of 2^m states that could be used to encode m bits with a good probability of recovery.

From a geometric perspective the problem can be viewed as one of packing subspaces. Each reflection R_j is about a certain subspace, projected to by $\frac{1}{2}(I + R_j)$. As explained in the previous section, the anticommutation condition implies that X_j and Z_j correspond to subspaces with all principal angles $\pi/4$, while the approximate commutation condition $\|[S_i, T_j]\| \leq \epsilon$ translates into the corresponding subspaces making principal angles close to 0 or $\pi/2$. By analogy to the problem of packing nearly orthogonal unit vectors[2] one might guess that as long as ϵ is not required to go to 0 too fast with n, m can be exponential in n.

The results in this section demonstrate that the geometric intuition is more accurate. Theorem 3.2 shows that for sufficiently small ϵ (inverse linear in n), no more than $m \leq n$

[2] For vector packing upper bounds on m, see, e.g., [7], [2, Lemma 9.1], [15].

ϵ-overlapping qubits can fit in 2^n dimensions. In contrast, Theorem 3.1 shows that as long as $\epsilon = \Omega(1)$, m can be exponential in n; more generally $m = \omega(n)$ for any $\epsilon = \omega(\sqrt{(\log n)/n})$. For the range of overlaps $1/n \lesssim \epsilon \lesssim \sqrt{(\log n)/n}$, we do not know whether strictly more than n qubits can be packed into 2^n dimensions.

3.1 Lower bound: packing exponentially many qubits in 2^n dimensions

We give a randomized construction that packs $m = e^{\Theta(n\epsilon^2)}$ qubits into 2^n dimensions. This beats the trivial $m = n$ for $\epsilon = \Omega(\sqrt{(\log n)/n})$, and is exponential in n for constant $\epsilon > 0$.

▶ **Theorem 3.1.** *There exist 2^n-dimensional reflections $X_1, Z_1, \ldots, X_m, Z_m$, for $m = e^{\Omega(n\epsilon^2)}$, such that $\{X_j, Z_j\} = 0$ and $\|[S_i, T_j]\| = O(\epsilon)$ for all $i \neq j$ and $S, T \in \{X, Z\}$.*

Proof. By the Johnson-Lindenstrauss Lemma [6, 5], $e^{n\epsilon^2/4}$ unit vectors can be chosen in $\mathbf{R}^{2n}$ so that for any pair $|u\rangle, |v\rangle$, $|\langle u|v\rangle| \leq \epsilon$. Collecting these vectors in triples, we obtain $m = \frac{1}{3}e^{n\epsilon^2/4}$ three-dimensional subspaces with the angles between any two in the range $[\frac{\pi}{2} - O(\epsilon), \frac{\pi}{2}]$. Let $\{|e_j\rangle, |f_j\rangle, |g_j\rangle\}$, for $j \in [m]$, be orthonormal bases for the subspaces.

Let $C_1, \ldots, C_{2n}$ denote a 2^n-dimensional representation of the Clifford algebra, i.e., Hermitian matrices that satisfy $\{C_i, C_j\} = 2\delta_{ij}\mathbf{1}$. For each $j \in [m]$, let

$$E_j = \sum_k \langle k|e_j\rangle\, C_k \qquad F_j = \sum_k \langle k|f_j\rangle\, C_k \qquad G_j = \sum_k \langle k|g_j\rangle\, C_k \ .$$

Then it is easy to check that for distinct $S, T \in \{E, F, G\}$, $\{S_j, T_j\} = 0$ and $\|\{S_i, T_j\}\| = O(\epsilon)$ for $i \neq j$. Let $X_j = iE_jF_j$ and $Z_j = iE_jG_j$; these matrices are Hermitian, square to $\mathbf{1}$, and anti-commute. Moreover, for $i \neq j$ and $S, T \in \{X, Z\}$, we have $\|[S_i, T_j]\| = O(\epsilon)$. ◀

Appendix A gives an alternative proof of Theorem 3.1 using the exterior algebra.

3.2 Upper bound: Separating overlapping qubit operators

We provide two different methods for creating independent qubits from partially overlapping qubits. The first argument, given in Section 3.2.1, performs a careful analysis of a sequential block-diagonalization procedure. The second argument, in Section 3.2.3, is simpler but requires the introduction of a larger Hilbert space in which to define the approximating operators.

3.2.1 Separating nearly commuting projections

We first consider the case of separating projections that nearly commute pairwise.

▶ **Theorem 3.2.** *Let $P_1, \ldots, P_n$ be projections on a finite-dimensional Hilbert space such that for some $\epsilon \leq \frac{1}{32n}$,*

$$\|[P_i, P_j]\| \leq \epsilon \qquad \text{for all } i, j.$$

Then there exist projections $Q_1, \ldots, Q_n$ with, for all i, j,

$$[Q_i, Q_j] = 0$$
$$\|P_i - Q_i\| \leq 8n\epsilon \ .$$

The bound in Theorem 3.2 is nearly tight; see Lemma 3.9 below.

The proof of the theorem is constructive. It uses two basic operations, that we analyze with two lemmas. First we block-diagonalize operators with respect to a projection Q so that they commute with Q. The first lemma bounds how block-diagonalizing two operators affects their commutator.

▶ **Lemma 3.3.** *Let Q be a projection, and for operators P_i, $i = 1, 2$, let $P_i' = QP_iQ + (\mathbf{1} - Q)P_i(\mathbf{1} - Q)$. Then $[Q, P_i'] = 0$, $\|P_i' - P_i\| = \|[Q, P_i]\|$, and*

$$\|[P_1', P_2']\| \leq \|[P_1, P_2]\| + 2\|[Q, P_1]\| \cdot \|[Q, P_2]\| \ .$$

Proof. Work in a basis in which Q is diagonal: $Q = \left(\begin{smallmatrix} \mathbf{1} & 0 \\ 0 & 0 \end{smallmatrix}\right)$. Then $P_i = \left(\begin{smallmatrix} A_i & B_i \\ C_i & D_i \end{smallmatrix}\right)$ and $P_i' = \left(\begin{smallmatrix} A_i & 0 \\ 0 & D_i \end{smallmatrix}\right)$. As $[Q, P_i] = \left(\begin{smallmatrix} 0 & B_i \\ -C_i & 0 \end{smallmatrix}\right)$, $\|P_i' - P_i\| = \max\{\|B_i\|, \|C_i\|\} = \|[Q, P_i]\|$. We also compute

$$[P_1, P_2] = \begin{pmatrix} [A_1, A_2] + B_1C_2 - B_2C_1 & A_1B_2 + B_1D_2 - A_2B_1 - B_2D_1 \\ C_1A_2 + D_1C_2 - C_2A_1 - D_2C_1 & [D_1, D_2] + C_1B_2 - C_2B_1 \end{pmatrix} \ .$$

Each diagonal block in $[P_1, P_2]$ above, $Q[P_1, P_2]Q$ and $(\mathbf{1} - Q)[P_1, P_2](\mathbf{1} - Q)$, must have norm at most $\|[P_1, P_2]\|$. The claimed bound for $\|[P_1', P_2']\| = \max\{\|[A_1, A_2]\|, \|[D_1, D_2]\|\}$ follows. ◀

When one block-diagonalizes a projection, the result might not be a projection. The second basic operation consists in rounding the eigenvalues to the closest integer, 0 or 1. The second lemma bounds how this affects the commutator with another operator.

▶ **Lemma 3.4.** *Let Q be a projection and Q' Hermitian with $[Q, Q'] = 0$ and $\|Q - Q'\| < 1/2$. Then for any Hermitian P,*

$$\|[Q, P]\| \leq \frac{\|[Q', P]\|}{1 - 2\|Q - Q'\|} \ .$$

This bound can be much stronger than the trivial $\|[Q, P]\| \leq \|[Q', P]\| + 2\|P\|\|Q - Q'\|$.[3] It follows by substituting $A = \left(\begin{smallmatrix} 0 & P(2Q-\mathbf{1}) \\ (2Q-\mathbf{1})P & 0 \end{smallmatrix}\right)$, $B = \left(\begin{smallmatrix} 0 & (2Q-\mathbf{1})P \\ P(2Q-\mathbf{1}) & 0 \end{smallmatrix}\right)$ and $\Gamma = |2Q' - \mathbf{1}| \oplus |2Q' - \mathbf{1}|$ into the following theorem, and using $|2Q' - \mathbf{1}|(2Q - \mathbf{1}) = (2Q - \mathbf{1})|2Q' - \mathbf{1}| = 2Q' - \mathbf{1}$.

▶ **Theorem 3.5** ([4, Theorem 1])**.** *If A and B are Hermitian, and $\Gamma \succ 0$, then*

$$\|A - B\| \leq \|\Gamma^{-1}\| \cdot \|A\Gamma - \Gamma B\| \ .$$

Proof of Theorem 3.2. We proceed inductively. The induction hypothesis is that we have defined $Q_1, \ldots, Q_k, P_{k+1}^{(k)}, \ldots, P_n^{(k)}$ such that

- $0 \preceq P_j^{(k)} \preceq \mathbf{1}$, $\|P_j^{(k)} - P_j\| \leq \delta_k$, $\|[P_i^{(k)}, P_j^{(k)}]\| \leq \epsilon_k$.
- $Q_1, \ldots, Q_k$ are projections, commuting with each other and all $P_j^{(k)}$, with $\|P_k - Q_k\| \leq 2\delta_{k-1}$.

[3] For $P \succeq 0$, trivially $\|[Q, P]\| \leq \|[Q', P]\| + \|[Q - Q', P - \frac{\|P\|}{2}\mathbf{1}]\| \leq \|[Q', P]\| + \|P\|\|Q - Q'\|$, but Lemma 3.4 is still stronger.

For the base case, $\delta_0 = 0$ and $\epsilon_0 = \epsilon$.

In the induction step, we let Q_{k+1} be the projection formed by rounding $P_{k+1}^{(k)}$'s eigenvalues to 0 or 1, and define $P_{k+2}^{(k+1)}, \ldots, P_n^{(k+1)}$ by block-diagonalizing the $P_j^{(k)}$ operators with respect to Q_{k+1}:

$$P_j^{(k+1)} = Q_{k+1} P_j^{(k)} Q_{k+1} + (\mathbf{1} - Q_{k+1}) P_j^{(k)} (\mathbf{1} - Q_{k+1}) \ .$$

Indeed, then $\|Q_{k+1} - P_{k+1}\| \leq \|P_{k+1}^{(k)} - P_{k+1}\| + \|Q_{k+1} - P_{k+1}^{(k)}\| \leq 2\delta_k$. Also, $0 \preceq P_j^{(k+1)} \preceq \mathbf{1}$. Using Lemma 3.3, we compute

$$\begin{aligned} \|P_j^{(k+1)} - P_j\| &\leq \|P_j^{(k)} - P_j\| + \|P_j^{(k+1)} - P_j^{(k)}\| \\ &\leq \delta_k + \|[Q_{k+1}, P_j^{(k)}]\| \\ \|[P_i^{(k+1)}, P_j^{(k+1)}]\| &\leq \|[P_i^{(k)}, P_j^{(k)}]\| + 2\|[Q_{k+1}, P_i^{(k)}]\| \cdot \|[Q_{k+1}, P_j^{(k)}]\| \ . \end{aligned}$$

Thus we may take $\delta_{k+1} = \delta_k + \max_j \|[Q_{k+1}, P_j^{(k)}]\|$ and $\epsilon_{k+1} = \epsilon_k + 2\max_j \|[Q_{k+1}, P_j^{(k)}]\|^2$. It remains to bound $\max_j \|[Q_{k+1}, P_j^{(k)}]\|$.

The naive bound $\|[Q_{k+1}, P_j^{(k)}]\| \leq \|[P_{k+1}^{(k)}, P_j^{(k)}]\| + 2\|Q_{k+1} - P_{k+1}^{(k)}\| \leq \epsilon_k + 2\delta_k$ is no good, as it allows the errors to grow exponentially with k. Instead, applying Lemma 3.4 gives

$$\left\|[Q_{k+1}, P_j^{(k)}]\right\| \leq \frac{\epsilon_k}{1 - 2\delta_k} \ .$$

Provided that all $\epsilon_k \leq 2\epsilon$ and $\delta_k \leq 1/4$, $(1 - 2\delta_k)^{-1} \leq 2$, and we obtain the recursions

$$\begin{aligned} \delta_{k+1} &\leq \delta_k + 2\epsilon_k \leq \delta_k + 4\epsilon \\ \epsilon_{k+1} &\leq \epsilon_k + 8\epsilon_k^2 \leq \epsilon_k + 32\epsilon^2 \ . \end{aligned}$$

Thus $\delta_{k+1} \leq 4(k+1)\epsilon$ and $\epsilon_{k+1} \leq \epsilon + 32k\epsilon^2$. Given $\epsilon \leq \frac{1}{32n}$, indeed $\epsilon_k \leq 2\epsilon$ and $\delta_k \leq 1/4$. ◀

3.2.2 Separating partially overlapping qubits

The following theorem is an extension of Theorem 3.2 which allows us to separate ϵ-overlapping qubits.

▶ **Theorem 3.6.** *Let $X_1, Z_1, \ldots, X_n, Z_n$ be Hermitian matrices each having eigenvalues in the range $[-1, -1+\epsilon] \cup [1-\epsilon, 1]$, and satisfying $\|\{X_j, Z_j\}\| \leq \epsilon$ and $\|[S_i, T_j]\| \leq \epsilon$ for all $i \neq j$ and $S, T \in \{X, Z\}$. Assume $\epsilon/(1-\epsilon)^2 \leq \frac{1}{64n}$. Then there exist reflections $X_1', Z_1', \ldots, X_n', Z_n'$ with $\{X_j', Z_j'\} = 0$, and $[S_i', T_j'] = 0$ and $\|S_j' - S_j\| \leq 4n\epsilon/(1-\epsilon)^2 + \epsilon$ for all $i \neq j$ and $S, T \in \{X, Z\}$.*

Proof. Let $\mathcal{H}$ be the finite-dimensional Hilbert space on which the matrices act. Introduce n additional qubits, and on $(\mathbf{C}^2)^{\otimes n} \otimes \mathcal{H}$, define

$$\begin{aligned} R_{2j-1}' &= \sigma_j^x \otimes X_j \\ R_{2j}' &= \sigma_j^z \otimes Z_j \ , \end{aligned}$$

for $j = 1, \ldots, n$, where σ_j^x and σ_j^z are the standard Pauli operators acting on the jth added qubit.

For Pauli operators σ and τ,

$$[\sigma \otimes A, \tau \otimes B] = \begin{cases} (\sigma\tau) \otimes [A, B] & \text{if } [\sigma, \tau] = 0 \\ (\sigma\tau) \otimes \{A, B\} & \text{if } \{\sigma, \tau\} = 0 \end{cases} \ .$$

Thus for all i, j,

$$\|[R'_i, R'_j]\| \leq \epsilon \ .$$

Define reflections $R_1, \ldots, R_{2n}$ by rounding to ± 1 the eigenvalues of each of $R'_1, \ldots, R'_{2n}$. The operators R_j still have the form (Pauli) $\otimes$ (Reflection). By Theorem 3.5,

$$\|[R_i, R_j]\| \leq \frac{1}{(1-\epsilon)^2}\epsilon \ .$$

Define projections $P_1, \ldots, P_{2n}$ by $P_j = \frac{1}{2}(\mathbf{1} + R_j)$. Then

$$\begin{aligned}\|[P_i, P_j]\| &= \tfrac{1}{4}\|[R_i, R_j]\| \\ &\leq \frac{1}{4}\frac{1}{(1-\epsilon)^2}\epsilon \ .\end{aligned}$$

Applying Theorem 3.2 for separating projections yields projections $Q_1, \ldots, Q_{2n}$ satisfying $[Q_i, Q_j] = 0$ and

$$\|Q_j - P_j\| \leq 8 \cdot (2n) \cdot \frac{1}{4}\frac{1}{(1-\epsilon)^2}\epsilon = \frac{4n\epsilon}{(1-\epsilon)^2} \ ,$$

provided that $\epsilon/(1-\epsilon)^2 \leq 1/(64n)$.

We claim that the reflections $2Q_{2j-1} - \mathbf{1}$ and $2Q_{2j} - \mathbf{1}$ still have the form $\sigma_j^x \otimes X'_j$ and $\sigma_j^z \otimes Z'_j$, resepectively, for reflections X'_j and Z'_j on $\mathcal{H}$. Indeed, the proof of the projections separation theorem, Theorem 3.2, involved two basic operations:

1. Block-diagonalizing an operator A with respect to a reflection R:

$$\begin{aligned}A &\to \tfrac{1}{2}(\mathbf{1} + R)A\tfrac{1}{2}(\mathbf{1} + R) + \tfrac{1}{2}(\mathbf{1} - R)A\tfrac{1}{2}(\mathbf{1} - R) \\ &= \frac{1}{2}(A + RAR) \ .\end{aligned}$$

2. Rounding the eigenvalues of a Hermitian operator A to ± 1.

Observe that if $A = \sigma \otimes A'$ for a Pauli σ, and $R = \tau \otimes R'$ for a Pauli τ, then both of these basic operations result in an operator $\sigma \otimes A''$, for the same Pauli σ.

Thus indeed $\{X'_j, Z'_j\} = 0$ and $[S'_i, T'_j] = 0$ for $i \neq j$ and $S, T \in \{X, Z\}$. Also $\|Q_j - P_j\| \leq 4n\epsilon/(1-\epsilon)^2$ implies

$$\begin{aligned}\|S'_j - S_j\| &\leq 2\|Q_j - P_j\| + \|R'_j - R_j\| \\ &\leq \frac{8n\epsilon}{(1-\epsilon)^2} + \epsilon \ .\end{aligned}$$

◀

Since Theorem 3.6 yields n qubits in tensor product, the dimension of the ambient space $\mathcal{H}$ must be at least 2^n. Rephrasing this, we obtain:

▶ **Corollary 3.7.** *In 2^n dimensions, at most n qubits can be placed with pairwise "overlaps" $\|[S_i, T_j]\| \leq \epsilon$, if $\epsilon/(1-\epsilon)^2 \leq 1/(64n)$.*

3.2.3 SWAP-based argument

If we are willing to work in a larger space, then there is a simpler argument for moving overlapping qubits into tensor product. Instead of repeatedly block-diagonalizing operators and rounding their eigenvalues to ± 1, as in Theorem 3.6, we can swap in fresh qubits to enforce a tensor-product structure. We will show:

▶ **Theorem 3.8.** *Let $X_1, Z_1, \ldots, X_n, Z_n$ be reflections on $\mathcal{H}$, satisfying $\{X_j, Z_j\} = 0$ and $\|[S_i, T_j]\| \leq \epsilon$ for all $i \neq j$ and $S, T \in \{X, Z\}$. Extend these operators by the identity to act on $\mathcal{H} \otimes (\mathbf{C}^2)^{\otimes n}$.*

Then there exist reflections $X'_1, Z'_1, \ldots, X'_n, Z'_n$ on $\mathcal{H} \otimes (\mathbf{C}^2)^{\otimes n}$, with $\{X'_j, Z'_j\} = 0$, $[S'_i, T'_j] = 0$ and $\|S'_j - S_j\| \leq 2n\epsilon$.

Proof. For $j \in [n]$, let $\mathcal{S}_j = \frac{1}{2}\big(\mathbf{1} \otimes \mathbf{1} + X_j \otimes \sigma_j^x + Z_j \otimes \sigma_j^z + i(X_j Z_j) \otimes \sigma_j^y\big)$. Acting on $\mathcal{H} \otimes (\mathbf{C}^2)^{\otimes n}$, $\mathcal{S}_j$ swaps the jth added $\mathbf{C}^2$ register with the qubit defined by X_j, Z_j.

For $T \in \{X, Z\}$ and $i \in \{1, \ldots, j\}$ define

$$T_j^{(i)} = (\mathcal{S}_1 \cdots \mathcal{S}_{i-1})\, T_j\, (\mathcal{S}_{i-1} \cdots \mathcal{S}_1) \ .$$

Let $T'_j = T_j^{(j)} = (\mathcal{S}_1 \cdots \mathcal{S}_{j-1}) T_j (\mathcal{S}_{j-1} \cdots \mathcal{S}_1)$.

Then for $i < j$, $\|[S'_i, T'_j]\| = \|[S_i, \mathcal{S}_i \cdots \mathcal{S}_{j-1} T_j \mathcal{S}_{j-1} \cdots \mathcal{S}_i]\|$. This is 0, since for any operator A that is the identity on the ith added $\mathbf{C}^2$ register, $[S_i, \mathcal{S}_i A \mathcal{S}_i] = 0$.

Furthermore,

$$\begin{aligned}
\|T'_j - T_j\| &\leq \sum_{i=1}^{j-1} \|T_j^{(i+1)} - T_j^{(i)}\| \\
&= \sum_{i=1}^{j-1} \|\mathcal{S}_i T_j \mathcal{S}_i - T_j\| \\
&= \sum_{i=1}^{j-1} \|[\mathcal{S}_i, T_j]\| \\
&\leq \frac{1}{2} \sum_{i=1}^{j-1} \big(\|[X_i, T_j]\| + \|[Z_i, T_j]\| + \|[X_i Z_i, T_j]\|\big) \\
&\leq 2\epsilon(j-1) \ .
\end{aligned}$$

◀

Since Theorem 3.8 works in the larger space $\mathcal{H} \otimes (\mathbf{C}^2)^{\otimes n}$, unlike Theorem 3.6 it does not give an upper bound on the number of nearly independent qubits that can be packed into $\mathcal{H}$.

3.2.4 Lower bound: Sometimes $\Omega(n\epsilon)$ movement is necessary

Theorem 3.6 shows that n qubits with pairwise "overlaps" at most ϵ can be separated into tensor product by moving each qubit $O(n\epsilon)$ in operator norm. Is the loss of a factor of n necessary? The following example shows that our bound is essentially tight.

▶ **Lemma 3.9.** *For any integer n, and any $\epsilon \in [0, \pi/n^2]$, there exist $2n$ qubits $X_1, Z_1, \ldots, X_{2n}, Z_{2n}$ in $(\mathbf{C}^2)^{\otimes(2n)}$ such that $\|[S_i, T_j]\| \leq \epsilon$ for all $i \neq j$ and $S, T \in \{X, Z\}$ but such that for any* independent *qubits $X'_1, Z'_1, \ldots, X'_{2n}, Z'_{2n}$ (with $[S'_i, T'_j] = 0$ for $i \neq j$),*

$$\max_{\substack{1 \leq j \leq 2n \\ S \in \{X,Z\}}} \big\|S_j - S'_j\big\| \geq \frac{n\epsilon}{2\pi} \ .$$

Proof. Construct qubits X_j, Z_j as the standard qubits, except with the second n qubit operators perturbed by the Hamiltonian

$$H = \tfrac{1}{4}(\sigma_1^z + \cdots + \sigma_n^z)(\sigma_{n+1}^z + \cdots + \sigma_{2n}^z) \ .$$

That is, $X_j = \sigma_j^x$, $Z_j = \sigma_j^z$ for $j \leq n$, and $X_j = e^{i\epsilon H} \sigma_j^x e^{-i\epsilon H}$, $Z_j = e^{i\epsilon H} \sigma_j^z e^{-i\epsilon H} = \sigma_j^z$ for $j > n$. Then if $j, k \leq n$ or $j, k > n$, the operators for qubits j and k commute. If $j \leq n < k$,

then the operators for qubits j and k commute, except for X_j and X_k. We compute $\|[X_j, X_k]\| = \|X_j X_k X_j - X_k\| = \|e^{-i\epsilon H}\sigma_j^x e^{i\epsilon H}\sigma_k^x e^{-i\epsilon H}\sigma_j^x e^{i\epsilon H} - \sigma_k^x\| = \|e^{i\epsilon\sigma_j^z\sigma_k^z} - \mathbf{1}\| = |e^{i\epsilon} - 1| \leq \epsilon$.

Let $X'_1, \ldots, X'_{2n}$ be any pairwise commuting reflections. Let $J = \{1, \ldots, n\}$, $K = \{n+1, \ldots, 2n\}$. Let $X_J = \prod_{j\in J} X_j$, $X_K = \prod_{k\in K} X_k$. Similarly define X'_J, X'_K and σ_J^x, σ_K^x. Thus $X_J = \sigma_J^x$, $X_K = e^{i\epsilon H}\sigma_K^x e^{-i\epsilon H}$. In order to lower-bound $\max_j \|X_j - X'_j\|$, we study $\|[X_J, X_K]\| = \|(X_J X_K)^2 - \mathbf{1}\|$.

On one hand, since the X'_j operators commute, $(X'_J X'_K)^2 = \mathbf{1}$. By triangle inequalities, and using $\|X_j\| = \|X'_j\| = 1$ for all j, $\|X_J X_K - X'_J X'_K\| \leq \sum_j \|X_j - X'_j\|$, and hence

$$\|(X_J X_K)^2 - \mathbf{1}\| \leq 2\sum_j \|X'_j - X_j\| \leq 4n \cdot \max_j \|X'_j - X_j\| \ . \tag{1}$$

On the other hand,

$$\begin{aligned}(X_J X_K)^2 &= \sigma_J^x\big(e^{i\epsilon H}\sigma_K^x\, e^{-i\epsilon H}\big)\sigma_J^x\big(e^{i\epsilon H}\sigma_K^x\, e^{-i\epsilon H}\big)\\ &= e^{-i\epsilon H}\sigma_K^x\, e^{2i\epsilon H}\sigma_K^x\, e^{-i\epsilon H}\\ &= e^{-4i\epsilon H}\ .\end{aligned}$$

Since $\|H\| = n^2/4$, provided that $n^2\epsilon \leq \pi$ it holds that

$$\big\|(X_J X_K)^2 - \mathbf{1}\big\| = |e^{in^2\epsilon} - 1| \geq \frac{2}{\pi}\cdot n^2\epsilon\ . \tag{2}$$

Combining the bounds (1) and (2) gives $\frac{2}{\pi}n^2\epsilon \leq \|(X_J X_K)^2 - \mathbf{1}\| \leq 4n \cdot \max_j \|X'_j - X_j\|$, or $\max_j \|X'_j - X_j\| \geq n\epsilon/(2\pi)$. ◀

4 State-dependent qubit separation

A problem with both Theorem 3.6 and Theorem 3.8 is that they might be difficult to apply to real experimental systems. This is because it is difficult to establish the assumption of qubits nearly in tensor product, $\|[S_i, T_j]\| \leq \epsilon$ for $i \neq j$ and $S, T \in \{X, Z\}$. In addition to the operators, a physical system involves an underlying state $|\psi\rangle$. The operators can be understood only in terms of their effects on $|\psi\rangle$. Consider for example a Hilbert space that splits as $\mathcal{H} \oplus \mathcal{H}'$, where $|\psi\rangle$ is supported only on $\mathcal{H}$ and available operators leave $\mathcal{H}$ invariant. Then there is no experimental way to fathom the operators' behavior, e.g., their commutation relationships, on $\mathcal{H}'$. Theorems 3.6 and 3.8 cannot be applied. This example might not seem so troubling, because we can simply restrict everything to $\mathcal{H}$; but it becomes more problematic if $|\psi\rangle$, say, has nonzero but very small support on $\mathcal{H}'$.

We would like qubit-separation theorems that have experimentally accessible assumptions. In particular, the theorems' assumptions should be stated relative to the system's state $|\psi\rangle$. For example, in Theorems 3.6 and 3.8 we might loosen the assumption $\|[S_i, T_j]\| \leq \epsilon$ for $i \neq j$ to be only $\|[S_i, T_j]|\psi\rangle\| \leq \epsilon$. Naturally, the conclusions will have to be correspondingly weakened. In the above example with $\mathcal{H} \oplus \mathcal{H}'$, if the reflections are far from commuting on $\mathcal{H}'$ then we cannot hope to find nearby commuting operators, $\|S'_j - S_j\| \approx 0$; but perhaps we can get $\|(S'_j - S_j)|\psi\rangle\| \approx 0$.

In order to extend our results to experimental systems we proceed in three steps.

1. First, in Section 4.1 below, we give a protocol that can be used to test if two reflections, S and T, are close to commuting on a state $|\psi\rangle$: $[S, T]|\psi\rangle \approx 0$. The protocol is very simple:

measure S, measure T, then measure S again. If S and T commute on $|\psi\rangle$, then the two S measurements will give the same result; and, intuitively, when they do not commute measuring T will disturb the state and make it less likely to get the same S result.

2. However, in Section 4.2, we show that the condition $[S_i, T_j]|\psi\rangle \approx 0$ for operators on different qubits is not sufficient to establish that there are nearby independent qubits $X'_1, Z'_1, \ldots, X'_n, Z'_n$. In fact, we give an explicit construction of a state $|\psi\rangle$ and n qubit operators $X_1, Z_1, \ldots, X_n, Z_n$ in $< n^2$ dimensions such that for $i \neq j$, $[S_i, T_j]|\psi\rangle = 0$ precisely. Since $n^2 \leq 2^n$ for $n \geq 4$, the dimension of the space is not sufficient to fit n independent qubits.

   (We also show why the basic induction argument used to prove Theorem 3.6 fails when errors are measured relative to a state $|\psi\rangle$. The errors accumulate too rapidly, leading to an exponential dependence on n, instead of polynomial.)

3. We remedy this problem in Section 4.3 with a more advanced testing protocol. Intuitively, the improved protocol tests not just pairwise commutation relationships, such as $S_i T_j |\psi\rangle \approx T_j S_i |\psi\rangle$, but also higher-order relationships such as $S_i T_j U_k |\psi\rangle \approx U_k T_j S_i |\psi\rangle$. The protocol is still quite simple, though. Basically, measure all the qubit operators in order (either $X_1, Z_1, X_2, Z_2, \ldots$ or $Z_1, X_1, Z_2, X_2, \ldots$), then go back and measure a random qubit operator (Z_j or X_j, respectively), and verify that the measurement result is unchanged. We show that if the protocol accepts with probability $1 - \epsilon$, then the qubit operators "simulate" n independent qubit operators in a certain sense. In particular, as a corollary, the system's dimension must be at least $(1 - O(n^2\epsilon))2^n$.

   The dimension bound is not fully satisfactory. A 2^n lower bound would be preferable. However, speculatively, the simulation statement might be strong enough to form the foundation for an analysis that the system can be used as an n-qubit quantum computer. Such an extension is nontrivial, though, and we leave it to future work.

4.1 Protocol for testing state-dependent commutation

We present a protocol that can be used to test whether two reflections approximately commute on a given state.

▶ **Theorem 4.1.** *Let S and T be reflections, acting on a state $|\psi\rangle$. Consider the following protocol:*

1. *Measure S.*
2. *Measure T, but ignore the result.*
3. *Measure S again. Accept if the result is unchanged.*

Then the probability of accepting is given by

$$\Pr[\text{accept}] = 1 - \tfrac{1}{8}\big\|[S,T]|\psi\rangle\big\|^2 \ .$$

Proof. For $a, b \in \{0,1\}$, let $S_a = \frac{1}{2}(\mathbf{1} + (-1)^a S)$ and $T_b = \frac{1}{2}(\mathbf{1} + (-1)^b T)$. Then since $[S, T_0] = -[S, T_1] = \frac{1}{2}[S, T]$,

$$\begin{aligned}\|[S,T]|\Psi\rangle\|^2 &= 2\big(\|[S,T_0]|\psi\rangle\|^2 + \|[S,T_1]|\psi\rangle\|^2\big) \\ &= \sum_{a,b} \big\|S_a[S,T_b]|\psi\rangle\big\|^2 \ ,\end{aligned}$$

where we have used $\||\phi\rangle\|^2 = \|S_0|\phi\rangle\|^2 + \|S_1|\phi\rangle\|^2$ for any $|\phi\rangle$. Then from $S_a S = S S_a = (-1)^a S_a$, we find $S_a[S, T_b] = S_a[S, T_b](S_0 + S_1) = 2(-1)^a S_a T S_{\bar{a}}$, so

$$\begin{aligned}\|[S,T]|\psi\rangle\|^2 &= 8\sum_{a,b}\left\|S_a T_b S_{\bar{a}}|\psi\rangle\right\|^2 \\ &= 8\left(1 - \Pr[\text{accept}]\right) \ .\end{aligned}$$

◀

4.2 Qubits that commute on a state need not be close to independent qubits

In the projection separating argument of Theorem 3.2, the key observation was that for projections P, Q, R with $\|[P,Q]\|, \|[P,R]\| \leq \delta$ and $\|[Q,R]\| \leq \epsilon$, if Q and R are both block-diagonalized with respect to P then the results still nearly commute:

$$\left\|\left[PQP + (\mathbf{1}-P)Q(\mathbf{1}-P), PRP + (\mathbf{1}-P)R(\mathbf{1}-P)\right]\right\| \leq \epsilon + 2\delta^2 \ .$$

The quadratic dependence on δ meant that errors did not accumulate badly through the induction.

Here is a counterexample showing that errors *can* accumulate badly in block diagonalization if we measure errors relative to a state $|\psi\rangle$, using $\|[P,Q]|\psi\rangle\|$. Define P, Q, R and $|\psi\rangle$ as

$$P = \begin{pmatrix} 1 & 0 & 0 & \delta \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ \delta & 0 & 0 & 0 \end{pmatrix} \qquad Q = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 1/2 & 1/2 \end{pmatrix} \qquad |\psi\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \ . \tag{3}$$

Then P, Q and R are projections (up to second order in δ for P), with $\|[P,Q]|\psi\rangle\|, \|[P,R]|\psi\rangle\| = O(\delta)$, $[Q,R]|\psi\rangle = 0$, and yet

$$\left\|\left[PQP + (\mathbf{1}-P)Q(\mathbf{1}-P), PRP + (\mathbf{1}-P)R(\mathbf{1}-P)\right]|\psi\rangle\right\| = \Omega(\delta) \ .$$

The idea is that Q and R commute on the first two dimensions, and are far from commuting on the last two dimensions; but this property is broken by the block diagonalization.

This example suggests that in a simple induction argument, starting with projections $P_1, \ldots, P_n$ having pairwise commutators $\|[P_i, P_j]|\psi\rangle\| \sim \epsilon$, after block-diagonalizing with respect to P_1, the errors can grow to $\sim 2\epsilon$, then to $\sim 4\epsilon$ after block-diagonalizing with respect to the new P_2, and so on; the errors potentially grow exponentially.

In fact, it is not only our *proof* of Theorems 3.2 and 3.6 that fails when errors are measured relative to a state $|\psi\rangle$. The theorems themselves fail, as shown by the following construction.

▶ **Lemma 4.2.** *For any n and $k \in [n]$, there exists a space $\mathcal{H}$ of dimension at most $1 + \sum_{j=0}^{k}\binom{n}{j}$, a vector $|\psi\rangle \in \mathcal{H}$ and n qubits X_j, Z_j such that*

$$S_{j_1}^{(1)} \cdots S_{j_k}^{(k)}|\psi\rangle = S_{j_{\sigma(1)}}^{\sigma(1)} \cdots S_{j_{\sigma(k)}}^{\sigma(k)}|\psi\rangle$$

for all distinct indices $j_1, \ldots, j_k \in [n]$, $S^{(1)}, \ldots, S^{(k)} \in \{X, Z\}$, and permutations σ of $[k]$.

In particular, for $k = 2$, the lemma places n qubits in $O(n^2)$ dimensions—for example, four qubits in 12 dimensions—such that $[S_i, T_j]|\psi\rangle = 0$ for all $i \neq j$ and $S, T \in \{X, Z\}$.

Protocol to test for n independent qubits

Let $|\psi\rangle \in \mathcal{H}$ be a state. Let $X_1, Z_1, \ldots, X_n, Z_n$ be qubit operators on $\mathcal{H}$, i.e., reflections satisfying $\{X_j, Z_j\} = 0$ for all j.

1. With equal probabilities 1/2, measure the reflections in order, either $X_1, Z_1, X_2, Z_2, \ldots$, or $Z_1, X_1, Z_2, X_2, \ldots$.
2. Pick a uniformly random index $j \in [n]$. If Z went second in step (1), then measure Z_j; and if X went second, then measure X_j. Accept if the result is unchanged from the operator's previous measurement. Otherwise reject.

Figure 2 Protocol to test for n independent qubits.

Proof. Let us begin by explaining the $n = 4$, $k = 2$ special case of the construction. Define $\mathcal{H}$ to have orthonormal basis $|0000\rangle, |1000\rangle, \ldots, |0001\rangle, |1100\rangle, \ldots, |0011\rangle, |d\rangle$, i.e., all n-bit strings of Hamming weight at most k, together with an additional vector $|d\rangle$. Let $|\psi\rangle = |0000\rangle$, and consider the following operators for the first qubit:

$$X_1 = \begin{array}{c} \\ 0000 \\ 1000 \\ 0100 \\ 0010 \\ 0001 \\ 1100 \\ 1010 \\ 1001 \\ 0110 \\ 0101 \\ 0011 \\ d \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 0000 & 1000 & 0100 & 0010 & 0001 & 1100 & 1010 & 1001 & 0110 & 0101 & 0011 & d \end{array} \\ \begin{pmatrix} 0 & 1 & & & & & & & & & & \\ 1 & 0 & & & & & & & & & & \\ & & 0 & & 1 & & & & & & & \\ & & & 0 & & 1 & & & & & & \\ & & & & 0 & & 1 & & & & & \\ & & 1 & & & 0 & & & & & & \\ & & & 1 & & & 0 & & & & & \\ & & & & 1 & & & 0 & & & & \\ & & & & & & & & 0 & 1 & & \\ & & & & & & & & 1 & 0 & & \\ & & & & & 0 & & & & & 0 & 1 \\ & & & & & & & & & & 1 & 0 \end{pmatrix} \end{array} \qquad Z_1 = \begin{array}{c} \\ 0000 \\ 1000 \\ 0100 \\ 0010 \\ 0001 \\ 1100 \\ 1010 \\ 1001 \\ 0110 \\ 0101 \\ 0011 \\ d \end{array} \begin{array}{c} \begin{array}{cccccccccccc} 0000 & 1000 & 0100 & 0010 & 0001 & 1100 & 1010 & 1001 & 0110 & 0101 & 0011 & d \end{array} \\ \begin{pmatrix} 1 & & & & & & & & & & & \\ & -1 & & & & & & & & & & \\ & & 1 & & & & & & & & & \\ & & & 1 & & & & & & & & \\ & & & & 1 & & & & & & & \\ & & & & & -1 & & & & & & \\ & & & & & & -1 & & & & & \\ & & & & & & & -1 & & & & \\ & & & & & & & & 1 & & & \\ & & & & & & & & & -1 & & \\ & & & & & 0 & & & & & 1 & \\ & & & & & & & & & & & -1 \end{pmatrix} \end{array} .$$

Unspecified matrix entries are 0. X_2 and Z_2 can be obtained from X_1 and Z_1 by switching the first and second bits in each basis element, leaving $|d\rangle$ alone; and similarly for X_3, Z_3 and X_4, Z_4. Then $P_i^2 = \mathbf{1}$, $\{X_i, Z_i\} = 0$ and $[P_i, Q_j]|\psi\rangle = 0$, for $i \neq j$ and $P, Q \in \{X, Z\}$.

The idea behind this construction is that X_j, Z_j act largely as the Pauli operators σ_j^x, σ_j^z. However, we have truncated the standard basis $|0000\rangle, \ldots, |1111\rangle$ for $(\mathbf{C}^2)^{\otimes 4}$ to include only strings of Hamming weight ≤ 2. Since applying σ_1^x to $|0110\rangle, |0101\rangle$ and $|0011\rangle$ would give strings of Hamming weight 3, we instead pair these dimensions up arbitrarily for X_1, and define Z_1 on them to make it anti-commute with X_1. The extra dimension $|d\rangle$ is needed to make the total dimension even.

It is straightforward to generalize the example: by truncating strings at Hamming weight k the same construction places n qubits in $\sum_{j=0}^{k} \binom{n}{j}$ dimensions (or one more if this dimension is odd), such that any combination of up to k qubit operators commute on $|\psi\rangle = |0^n\rangle$, e.g., if $k \geq 3$, $X_1 X_2 X_3 |\psi\rangle = X_3 X_2 X_1 |\psi\rangle = |1110^{n-3}\rangle$. ◀

4.3 Protocol to test for n independent qubits

The problem with the protocol in Theorem 4.1 is that it only tests commutation between pairs of operators on the state $|\psi\rangle$: $[S, T]|\psi\rangle \approx 0$. Lemma 4.2 shows that n qubits in only $O(n^2)$ dimensions can pass this test on every pair. The lemma furthermore suggests that any test involving qubit operator sequences of length $o(n)$ can be satisfied in dimension $2^{o(n)}$. Therefore, we need a protocol that has at least n steps.

Figure 2 gives our testing protocol. We argue that if the protocol accepts with high probability, then the n overlapping qubits X_j, Z_j are nearly equivalent to n independent qubits $\hat{X}_j, \hat{Z}_j$ in an enlarged space $\mathcal{H}' = \mathcal{H} \otimes (\mathbf{C}^2)^{\otimes 2n}$.

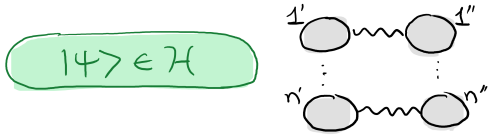


Figure 3 The state $|\Psi_0\rangle$ is given by $|\psi\rangle \otimes |\mathrm{EPR}\rangle^{\otimes n}$, where the EPR states are on qubits $1'$ and $1''$, $2'$ and $2''$, and so on. To get $|\Psi\rangle$, swap qubit j' with the qubit in $\mathcal{H}$ defined by X_j, Z_j, for $j = 1, \ldots, n$. Observe that starting from $|\psi\rangle$ and depolarizing the X_j, Z_j qubits, for $j = 1, \ldots, n$, is equivalent to tracing out all j' and j'' qubits from $|\Psi\rangle\langle\Psi|$.

▶ **Theorem 4.3.** *Consider the protocol of Figure 2. Assume the probability it accepts is at least* $1 - \epsilon$.

Let $|\mathrm{EPR}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. *Let* $|\Psi_0\rangle = |\psi\rangle \otimes |\mathrm{EPR}\rangle^{\otimes n} \in \mathcal{H}' = \mathcal{H} \otimes (\mathbf{C}^2)^{\otimes 2n}$, *and let* $|\Psi\rangle$ *be obtained from* $|\Psi_0\rangle$ *by swapping each qubit* X_j, Z_j *with the first half of one of the EPR states, in order* $j = 1, \ldots, n$. *(See Figure 3.) Then there exist* n *independent qubits, given by* $\hat{X}_1, \hat{Z}_1, \ldots, \hat{X}_n, \hat{Z}_n$, *on* $\mathcal{H}'$ *such that for any sequence of qubit operators* $U_{j_1}, \ldots, U_{j_k}$, *where* U_j *acts on the* X_j, Z_j *qubit and* $\|U_j\| \leq 1$,

$$\left\| U_{j_1} \cdots U_{j_k} |\Psi\rangle - \hat{U}_{j_1} \cdots \hat{U}_{j_k} |\Psi\rangle \right\| = O(kn\sqrt{\epsilon}) \ . \tag{4}$$

Here $\hat{U}_j$ *is the same operator as* U_j, *except acting on the* $\hat{X}_j, \hat{Z}_j$ *qubit. That is, if* U_j *has Pauli expansion* $U_j = \alpha_j \mathbf{1} + \beta_j X_j + \gamma_j Z_j + \delta_j (i X_j Z_j)$ *for scalars* $\alpha_j, \beta_j, \gamma_j, \delta_j$, *then* $\hat{U}_j = \alpha_j \mathbf{1} + \beta_j \hat{X}_j + \gamma_j \hat{Z}_j + \delta_j (i \hat{X}_j \hat{Z}_j)$.

Observe that if the X_j, Z_j qubits are independent of each other, then the measurements on different qubits commute, and so the protocol accepts with probability one. In that case, there is nothing to show. In general, however, measuring qubits $j+1, \ldots, n$ can disturb the last measurement on qubit j.

The EPR state appears in the conclusion of Theorem 4.3 even though it is not used in the testing protocol. Essentially this is because of the following two properties of $|\mathrm{EPR}\rangle$:

1. Depolarizing a qubit, i.e., replacing it with the maximally mixed state, is equivalent to swapping it with the first qubit of a fresh EPR state then tracing out the EPR state's registers.
2. For any 2×2 matrix M, $(I \otimes M)|\mathrm{EPR}\rangle = (M^T \otimes I)|\mathrm{EPR}\rangle$.

The second property is key in our analysis for algebraically manipulating operators to show approximate commutation. To see how, consider for example a state $|\phi\rangle$ that involves four qubits, labeled $1, 2, 1', 2'$, where the j' qubits do not overlap with any others. If $|\phi\rangle$ is close to an EPR state on qubits $(1, 1')$ and $(2, 2')$, then operators on qubits 1 and 2 necessarily nearly commute on $|\phi\rangle$:

$$\begin{aligned} U_1 V_2 |\phi\rangle &\approx U_1 V_{2'}^T |\phi\rangle = V_{2'}^T U_1 |\phi\rangle \\ &\approx V_{2'}^T U_{1'}^T |\phi\rangle = U_{1'}^T V_{2'}^T |\phi\rangle \\ &\approx U_{1'}^T V_2 |\phi\rangle = V_2 U_{1'}^T |\phi\rangle \\ &\approx V_2 U_1 |\phi\rangle \ . \end{aligned}$$

The trick is to pull operators from one side of an approximate EPR state to the other, commute them there, then pull them back.

Proof of Theorem 4.3. To analyze the protocol, we relate it to a separate protocol that is based on swapping qubits with halves of EPR states. Observe that measuring either X_i

then Z_i, or Z_i then X_i, and discarding the second measurement result, is equivalent to depolarizing the qubit. Depolarizing a qubit is equivalent to swapping it with one half of $|\mathrm{EPR}\rangle$ and tracing out the original EPR state's registers. Therefore, the protocol of Figure 2 accepts with the same probability as the following protocol:

1. Append to the system n EPR states, on qubits labeled $1', 1'', \ldots, n', n''$. Thus the system is in the state $|\Psi_0\rangle = |\psi\rangle \otimes |\mathrm{EPR}\rangle^{\otimes n} \in \mathcal{H} \otimes (\mathbf{C}^2_{1'} \otimes \mathbf{C}^2_{1''}) \otimes \cdots \otimes (\mathbf{C}^2_{n'} \otimes \mathbf{C}^2_{n''})$; see Figure 3.
2. For i from 1 up to n, swap the qubit defined by X_i, Z_i with the new qubit i'.
3. Pick a uniformly random index $j \in [n]$. With equal probabilities 1/2, measure either X_j and $\sigma^x_{j''}$, or Z_j and $\sigma^z_{j''}$. Accept if the measurement results are the same, both $+1$ or both -1.

Indeed, for $\alpha \in \{x, z\}$, measuring $\sigma^\alpha_{j''}$ at the end of the protocol is equivalent to measuring $\sigma^\alpha_{j'}$ at the start, which is also equivalent to measuring just after swapping with the X_j, Z_j qubit.

If the protocol accepts with probability $1-\epsilon$, then for probabilities ϵ_j satisfying $\epsilon = \frac{1}{n}\sum_j \epsilon_j$, we have $\min\{\|\frac{1}{2}(\mathbf{1} + X_j \otimes \sigma^x_{j''})|\Psi\rangle\|^2, \|\frac{1}{2}(\mathbf{1} + Z_j \otimes \sigma^z_{j''})|\Psi\rangle\|^2\} \geq 1 - 2\epsilon_j$, where $|\Psi\rangle$ is the state after the swap gates in step (2). In particular,

$$\max\left\{\left\|X_j \otimes \sigma^x_{j''}|\Psi\rangle - |\Psi\rangle\right\|, \left\|Z_j \otimes \sigma^z_{j''}|\Psi\rangle - |\Psi\rangle\right\|\right\} \leq 2\sqrt{2\epsilon_j} \ .$$

This implies that for any one-qubit operator U_j acting on the X_j, Z_j qubit, $U_j|\Psi\rangle \approx U^T_{j''}|\Psi\rangle$, where $U_{j''}$ is the same operator, but acting on the j'' qubit. More precisely, if $U_j = \alpha_j \mathbf{1} + \beta_j X_j + \gamma_j Z_j + \delta_j(iX_jZ_j)$ for complex scalars $\alpha_j, \beta_j, \gamma_j, \delta_j$, then $U^T_{j''} = \alpha_j \mathbf{1} + \beta_j \sigma^x_{j''} + \gamma_j \sigma^z_{j''} - \delta_j \sigma^y_{j''}$; and, since $\max\{|\alpha_j|, |\beta_j|, |\gamma_j|, |\delta_j|\} \leq \|U_j\|$,

$$\begin{aligned}\left\|(U_j - U^T_{j''})|\Psi\rangle\right\| &\leq (|\beta_j| + |\gamma_j| + 2|\delta_j|) \cdot 2\sqrt{2\epsilon_j} \\ &\leq 4\|U_j\| \cdot 2\sqrt{2\epsilon_j} \ .\end{aligned}$$

For each i, let $\mathcal{S}_i$ be the operator on that swaps the X_i, Z_i qubit with the new qubit i': $\mathcal{S}_i = \frac{1}{2}\big(\mathbf{1} + X_i \otimes \sigma^x_{i'} + Z_i \otimes \sigma^z_{i'} + i(X_iZ_i) \otimes \sigma^y_{i'}\big)$. For $i \leq j$, let $\mathcal{S}_{i,j} = \mathcal{S}_i\mathcal{S}_{i+1}\ldots\mathcal{S}_j$ and $\mathcal{S}_{j,i} = \mathcal{S}_j\mathcal{S}_{j-1}\ldots\mathcal{S}_i$. Thus $|\Psi\rangle = \mathcal{S}_{n,1}|\Psi_0\rangle$.

Let $\hat{P}_i = \mathcal{S}_{n,i+1}P_i\mathcal{S}_{i+1,n} = \mathcal{S}_{n,i}\sigma^P_{i'}\mathcal{S}_{i,n} = \mathcal{S}_{n,1}\sigma^P_{i'}\mathcal{S}_{1,n}$. As $[\sigma^P_{i'}, \sigma^Q_{j'}] = 0$ for $i \neq j$ and $P, Q \in \{X, Z\}$, so too $[\hat{P}_i, \hat{Q}_j] = 0$. Observe that

$$\hat{U}_j|\Psi\rangle = U^T_{j''}|\Psi\rangle \ , \tag{5}$$

since

$$\begin{aligned}\hat{U}_j\mathcal{S}_{n,1}|\Psi_0\rangle &= (\mathcal{S}_{n,1}U_{j'}\mathcal{S}_{1,n})\mathcal{S}_{n,1}|\Psi_0\rangle \\ &= \mathcal{S}_{n,1}U_{j'}|\Psi_0\rangle \\ &= \mathcal{S}_{n,1}U^T_{j''}|\Psi_0\rangle \ ,\end{aligned}$$

where the last equality is because $|\Psi_0\rangle$ includes an EPR state between qubits j' and j''. It follows that for any unitary U acting only on the X_j, Z_j qubit,

$$\left\|(U_j - \hat{U}_j)|\Psi\rangle\right\| \leq 8\sqrt{2\epsilon_j} \ . \tag{6}$$

Now consider a sequence of operators $U_{j_1}, \ldots, U_{j_k}$, where U_j acts on the X_j, Z_j qubit and

$\|U_j\| \le 1$. Then iterating $\hat{U}_j|\Psi\rangle = U_{j''}^T|\Psi\rangle$ gives

$$\begin{aligned}\hat{U}_{j_1}\cdots\hat{U}_{j_k}|\Psi\rangle &= \hat{U}_{j_1}\cdots\hat{U}_{j_{k-1}}U_{j_k''}^T|\Psi\rangle \\ &= U_{j_k''}^T\hat{U}_{j_1}\cdots\hat{U}_{j_{k-1}}|\Psi\rangle \\ &= \cdots \\ &= U_{j_k''}^T\cdots U_{j_1''}^T|\Psi\rangle \ .\end{aligned}$$

To continue, iterate on $U_j|\Psi\rangle \approx U_{j''}^T|\Psi\rangle$:

$$\begin{aligned}&\approx U_{j_1}U_{j_k''}^T\cdots U_{j_2''}^T|\Psi\rangle \\ &\approx \cdots \\ &\approx U_{j_1}\cdots U_{j_k}|\Psi\rangle \ .\end{aligned}$$

The overall error satisfies

$$\big\|U_{j_1}\cdots U_{j_k}|\Psi\rangle - \hat{U}_{j_1}\cdots\hat{U}_{j_k}|\Psi\rangle\big\| \le k\cdot 4\max\|U_{j_\ell}\|\cdot 2\sqrt{2\epsilon_{j_\ell}} = O(k\sqrt{n\epsilon}) \ . \qquad \blacktriangleleft$$

In Theorem 4.3, the definition of $|\Psi\rangle$ requires adding to $\mathcal{H}$ an additional ancilla register $(\mathbf{C}^2)^{\otimes 2n}$. It is therefore not clear that the theorem should imply an exponential lower bound on the dimension of $\mathcal{H}$. In fact, though, it does lower-bound $\dim\mathcal{H}$:

▶ **Corollary 4.4.** *If the protocol in Figure 2 accepts with probability at least $1-\epsilon$, then*

$$\dim\mathcal{H} \ge \big(1-O(n^2\epsilon)\big)\,2^n \ .$$

Proof. For $(a,b)\in\{0,1\}^n\times\{0,1\}^n$ let

$$|\Psi_{a,b}\rangle = \big(X_n^{a_n}Z_n^{b_n}\big)\cdots\big(X_1^{a_1}Z_1^{b_1}\big)|\Psi\rangle \ .$$

▶ Claim 4.5. The $|\Psi_{a,b}\rangle$ satisfy $\dim\mathrm{Span}\{|\Psi_{a,b}\rangle\} \ge \big(1-O(n^2\epsilon)\big)4^n$.

Proof. Let $B = \sum_{a,b}|\Psi_{a,b}\rangle\langle a,b|$. Adopt the notation from the proof of Theorem 4.3. For $k\in\{0,\ldots,n\}$ define $|\hat{\Psi}_{a,b}^{(k)}\rangle$ similarly to $|\Psi_{a,b}\rangle$, except using the operators $\hat{X}_j$ and $\hat{Z}_j$ in place of X_j and Z_j for $j\le k$. Thus $|\hat{\Psi}_{a,b}^{(0)}\rangle = |\Psi_{a,b}\rangle$. Let $|\hat{\Psi}_{a,b}\rangle = |\hat{\Psi}_{a,b}^{(n)}\rangle$ and define $\hat{B}$ as B using the $|\hat{\Psi}_{a,b}\rangle$ instead of $|\Psi_{a,b}\rangle$. Using the triangle inequality and $\|X_j\|, \|Z_j\| \le 1$,

$$\begin{aligned}\big\||\hat{\Psi}_{a,b}\rangle - |\Psi_{a,b}\rangle\big\| &\le \sum_{k=1}^n \big\||\hat{\Psi}_{a,b}^{(k)}\rangle - |\hat{\Psi}_{a,b}^{(k-1)}\rangle\big\| \\ &\le \sum_{k=1}^n \Big\|\big(\hat{X}_k^{a_k}\hat{Z}_k^{b_k} - X_k^{a_k}Z_k^{b_k}\big)\Big(\prod_{j<k}\hat{X}_j^{a_j}\hat{Z}_j^{b_j}\Big)|\Psi\rangle\Big\| \ . \qquad (7)\end{aligned}$$

By Eq. (5) from the proof of Theorem 4.3, $\hat{P}_j|\Psi\rangle = P_{j''}^T|\Psi\rangle$, where $P_{j''}$ acts only on the j'' ancilla qubit and therefore commutes with all Q_k and $\hat{Q}_k$. Thus for any $k\in[n]$,

$$\big(\hat{X}_k^{a_k}\hat{Z}_k^{b_k} - X_k^{a_k}Z_k^{b_k}\big)\Big(\prod_{j<k}\hat{X}_j^{a_j}\hat{Z}_j^{b_j}\Big)|\Psi\rangle = \Big(\prod_{j<k}\big(X_{j''}^{a_j}Z_{j''}^{b_j}\big)^T\Big)\big(\hat{X}_k^{a_k}\hat{Z}_k^{b_k} - X_k^{a_k}Z_k^{b_k}\big)|\Psi\rangle \ .$$

Thus starting from Eq. (7) and applying (6), we obtain the bound

$$\big\||\hat{\Psi}_{a,b}\rangle - |\Psi_{a,b}\rangle\big\| \le \sum_{k=1}^n 8\sqrt{2\epsilon_k} \ . \qquad (8)$$

Moreover, the $|\hat{\Psi}_{a,b}\rangle$ vectors are orthonormal:

$$\begin{aligned}\langle\hat{\Psi}_{a,b}|\hat{\Psi}_{c,d}\rangle &= \langle\Psi_0|\mathcal{S}_{1,n}\prod_{j=1}^{n}\big(\hat{Z}_j^{b_j}\hat{X}_j^{a_j+c_j}\hat{Z}_j^{d_j}\big)\mathcal{S}_{n,1}|\Psi_0\rangle \\ &= (-1)^{(a+c)\cdot b}\langle\mathrm{EPR}|^{\otimes n}\prod_{j=1}^{n}\big((\sigma_{j'}^x)^{a_j+c_j}(\sigma_{j'}^z)^{b_j+d_j}\big)|\mathrm{EPR}\rangle^{\otimes n} \\ &= \delta_{a,c}\delta_{b,d}\ .\end{aligned}$$

Therefore $\hat{B}$ is an isometry. Its singular values are 1 with multiplicity 4^n. Let $\lambda_1 \geq \cdots \geq \lambda_{4^n} \geq 0$ be the singular values of B. (Some λ_i may be zero.) Then, relating the singular values of B and $\hat{B}$ to the Frobenius norm of their difference,

$$\begin{aligned}\sum_i |\lambda_i - 1|^2 &\leq \|B - \hat{B}\|_F^2 \\ &= \sum_{a,b}\big\||\Psi_{a,b}\rangle - |\hat{\Psi}_{a,b}\rangle\big\|^2 \\ &\leq 4^n \cdot 128 \cdot n^2\epsilon\ ,\end{aligned}$$

where the last bound is by Eq. (8) and $\sum_k \epsilon_k = n\epsilon$. Since the left-hand side is at least $4^n - \mathrm{rank}(B)$, we obtain $\mathrm{rank}(B) \geq \big(1 - O(n^2\epsilon)\big)4^n$. ◀

Let $|\Psi\rangle$ have Schmidt decomposition $|\Psi\rangle = \sum_{i=1}^{d}\sqrt{p_i}|u_i\rangle \otimes |v_i\rangle$ across the partition $\mathcal{H}$, $(\mathbf{C}^2)^{\otimes 2n}$. Extend the set $\{|u_1\rangle, \ldots, |u_d\rangle\}$, if necessary, to form an orthonormal basis for $\mathcal{H}$. The vectors $|\Psi_{a,b}\rangle$ are obtained from $|\Psi\rangle$ by applying operators X_j, Z_j supported only on $\mathcal{H}$. Therefore, they lie in the span of $\{|u_i\rangle \otimes |v_j\rangle : i \in [\dim\mathcal{H}], j \in [d]\}$. In particular, $\dim \mathrm{Span}\{|\Psi_{a,b}\rangle\} \leq d\dim\mathcal{H} \leq (\dim\mathcal{H})^2$, as desired. ◀

▶ Remark. In Theorem 3.6, different qubits overlapping by $\epsilon = O(1/n)$ already implies $\dim\mathcal{H} \geq 2^n$. In contrast, in Corollary 4.4, ϵ must be exponentially small before $\dim\mathcal{H} \geq 2^n$ is required. Is this polynomial versus exponential separation a consequence of loose analysis, an inherent drawback of the protocol in Figure 2, or an inherent property of any efficient state-dependent qubit testing protocol?

The following example suggests at least that our analysis is not too loose. Let $\mathcal{H} = \mathrm{Span}\{|x\rangle : x \neq 0^n, 1^n\} \subset (\mathbf{C}^2)^{\otimes n}$. Define n qubits by $Z_j = \sigma_j^z|_{\mathcal{H}}$ and $X_j = \sigma_j^x|_{\mathcal{H}} + \sigma_j^x(|1^n\rangle\langle 0^n| + |0^n\rangle\langle 1^n|)\sigma_j^x$. That is, while σ_j^x maps the basis states $\sigma_j^x|0^n\rangle$ and $\sigma_j^x|1^n\rangle$ outside of $\mathcal{H}$, X_j instead maps them to each other. Even though $\dim\mathcal{H} = 2^n - 2 < 2^n$, it seems that these n qubits can pass our testing protocol with probability $1 - 1/\exp(n)$.[4]

Acknowledgements. We would like to thank Greg Kuperberg for helpful comments, particularly regarding the proof of Theorem 3.1.

References

1 Tameem Albash, Itay Hen, Federico M. Spedalieri, and Daniel A. Lidar. Reexamination of the evidence for entanglement in the D-Wave processor. *Phys. Rev. A*, 92:062328, 2015. doi:10.1103/PhysRevA.92.062328.

[4] A natural generalization of this construction removes all strings of Hamming weight $< t$ or $> n - t$, with $Z_j = \sigma_j^z|_{\mathcal{H}}$ and $X_j|x\rangle = \sigma_j^x|x\rangle$ except $X_j|x\rangle = |\overline{x}\rangle$ when $\sigma_j^x|x\rangle$ would cross the boundary. We omit the details.

2 Noga Alon. Problems and results in extremal combinatorics—I. *Discrete Mathematics*, 273(1-3):31–53, 2003. EuroComb'01. doi:10.1016/S0012-365X(03)00227-9.

3 László Babai and Katalin Friedl. Approximate representation theory of finite groups. In *Foundations of Computer Science, 1991. Proceedings., 32nd Annual Symposium on*, pages 733–742. IEEE, 1991. doi:10.1109/SFCS.1991.185442.

4 Rajendra Bhatia, Chandler Davis, and Fuad Kittaneh. Some inequalities for commutators and an application to spectral variation. *Aequationes Mathematicae*, 41(1):70–78, 1991. doi:10.1007/BF02227441.

5 Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures and Algorithms*, 22(1):60–65, 2003. doi:10.1002/rsa.10073.

6 William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conf. on modern analysis and probability, New Haven, CI, 1982*, volume 26 of *Contemporary Mathematics*, pages 189–206. Amer. Math. Soc., Providence, RI, 1984. doi:10.1090/conm/026/737400.

7 G. A. Kabatjanskiĭ and V. I. Levenšteĭn. Bounds for packings on the sphere and in space. *Problemy Peredači Informacii*, 14(1):3–25, 1978.

8 Jedrzej Kaniewski, Marco Tomamichel, and Stephanie Wehner. Entropic uncertainty from effective anticommutators. *Physical Review A*, 90(1):012332, 2014. doi:10.1103/PhysRevA.90.012332.

9 Greg Kuperberg. Personal communication, February 2014.

10 Terry A. Loring. C*-algebras generated by stable relations. *J. Functional Analysis*, 112(1):159–203, 1993. doi:10.1006/jfan.1993.1029.

11 Dominic Mayers and Andrew Yao. Quantum cryptography with imperfect apparatus. In *Proc. 39th IEEE FOCS*, pages 503–509, 1998. doi:10.1109/SFCS.1998.743501.

12 Matthew McKague, Tzyh Haur Yang, and Valerio Scarani. Robust self-testing of the singlet. *J. Phys. A: Math. Theor.*, 45:455304, 2012. doi:10.1088/1751-8113/45/45/455304.

13 Cristopher Moore and Alexander Russell. Approximate representations, approximate homomorphisms, and low-dimensional embeddings of groups. *SIAM Journal on Discrete Mathematics*, 29(1):182–197, 2015. doi:10.1137/140958578.

14 Ashwin Nayak. Optimal lower bounds for quantum automata and random access codes. In *Proc. 40th IEEE FOCS*, pages 369–376, 1999. doi:10.1109/SFFCS.1999.814608.

15 Terence Tao. A cheap version of the Kabatjanskii-Levenstein bound for almost orthogonal vectors, July 2013. URL: https://terrytao.wordpress.com/2013/07/18/a-cheap-version-of-the-kabatjanskii-levenstein-bound-for-almost-orthogonal-vectors/.

A Qubit packing using the exterior algebra

An alternative proof of Theorem 3.1 was suggested to the authors by Greg Kuperberg [9]. The rough idea is to begin by packing nearly orthogonal unit vectors in $\mathbf{R}^n$, then define qubits using fermion creation and annihilation operators on the 2^n-dimensional exterior algebra.

Proof of Theorem 3.1. By the Johnson-Lindenstrauss Lemma [6, 5], $e^{n\epsilon^2/8}$ unit vectors can be chosen in $\mathbf{R}^n$ so that for any pair $|u\rangle, |v\rangle$, $|\langle u|v\rangle| \leq \epsilon$. Pairing these vectors up arbitrarily, we obtain $m = \frac{1}{2}e^{n\epsilon^2/8}$ two-dimensional planes the angles between any two of which are in the range $(\frac{\pi}{2} - \epsilon, \frac{\pi}{2}]$.

If $|1\rangle, \ldots, |n\rangle$ is a basis for $\mathbf{R}^n$, let $\Lambda(\mathbf{R}^n)$ be the 2^n-dimensional exterior algebra, with basis $|i_1\rangle \wedge |i_2\rangle \wedge \ldots \wedge |i_k\rangle$ for $i_1, \ldots, i_k \in [n]$ and $k = 0, 1, \ldots, n$. For a unit vector $|v\rangle \in \mathbf{R}^n$ and $|w\rangle \in \Lambda(\mathbf{R}^n)$, define the fermion creation and annihilation operators

$$a_v^\dagger |w\rangle = |v\rangle \wedge |w\rangle$$
$$a_v |w\rangle = (\langle v| \otimes \mathbf{1})|w\rangle \ .$$

Observe that this definition is basis independent, in the sense that for any unitary R on $\mathbf{R}^n$,

$$a_{Rv}^\dagger \hat{R}|w\rangle = \hat{R} a_v^\dagger |w\rangle$$
$$a_{Rv} \hat{R}|w\rangle = \hat{R} a_v |w\rangle \ ,$$

where $\hat{R}(|v_1\rangle \wedge \cdots \wedge |v_k\rangle) = (R|v_1\rangle) \wedge \cdots \wedge (R|v_k\rangle)$.

If we choose a basis for $\mathbf{R}^n$ beginning with $|v\rangle$, then $a_v^\dagger a_v$ projects onto those basis terms in $\Lambda(\mathbf{R}^n)$ that include $|v\rangle$, while $a_v a_v^\dagger$ projects onto the complementary set of basis terms. Thus $a_v^\dagger a_v + a_v a_v^\dagger = \mathbf{1}$, while also $a_v^2 = (a_v^\dagger)^2 = 0$. Furthermore, if $|u\rangle$ is a unit vector perpendicular to $|v\rangle$, then the anticommutators satisfy $\{a_v, a_u\} = \{a_v^\dagger, a_u^\dagger\} = 0$, as $|u\rangle \wedge |v\rangle = -|v\rangle \wedge |u\rangle$, while if $|w\rangle$ has k terms,

$$\begin{aligned} a_u a_v^\dagger |w\rangle &= (\langle u| \otimes \mathbf{1})(|v\rangle \wedge |w\rangle) \\ &= (-1)^k (\langle u| \otimes \mathbf{1}|w\rangle \wedge |v\rangle \\ &= -a_v^\dagger a_u |w\rangle \ . \end{aligned}$$

Thus $\{a_u, a_v^\dagger\} = 0$.

Now for each of the m pairwise nearly orthogonal planes, let $\{|u_j\rangle, |v_j\rangle\}$ constitute an orthonormal basis. Define

$$X_j = (-a_{u_j} + a_{u_j}^\dagger)(a_{v_j} + a_{v_j}^\dagger)$$
$$Z_j = 2a_{v_j} a_{v_j}^\dagger - \mathbf{1} = a_{v_j} a_{v_j}^\dagger - a_{v_j}^\dagger a_{v_j} \ . \tag{9}$$

To understand this construction, observe that for orthonormal vectors $|u\rangle, |v\rangle \in \mathbf{R}^n$, and any $|w\rangle \in \Lambda(\mathbf{R}^n)$ with $a_u|w\rangle = a_v|w\rangle = 0$, the operators $a_u, a_u^\dagger, a_v, a_v^\dagger$ fix the subspace spanned by $|w\rangle, |v\rangle \wedge |w\rangle, |u\rangle \wedge |w\rangle, |u\rangle \wedge |v\rangle \wedge |w\rangle$. In this basis,

$$a_u = \begin{pmatrix} 0&0&1&0 \\ 0&0&0&1 \\ 0&0&0&0 \\ 0&0&0&0 \end{pmatrix} \qquad a_v = \begin{pmatrix} 0&1&0&0 \\ 0&0&0&0 \\ 0&0&0&-1 \\ 0&0&0&0 \end{pmatrix} \ .$$

Hence,

$$(-a_u + a_u^\dagger)(a_v + a_v^\dagger) = \begin{pmatrix} 0&0&0&1 \\ 0&0&1&0 \\ 0&1&0&0 \\ 1&0&0&0 \end{pmatrix} \qquad 2a_v a_v^\dagger - \mathbf{1} = \begin{pmatrix} 1&0&0&0 \\ 0&-1&0&0 \\ 0&0&1&0 \\ 0&0&0&-1 \end{pmatrix} \ .$$

The former matrix is $\sigma_X \otimes \sigma_X$, and the latter matrix is $I \otimes \sigma_Z$, where σ_X, σ_Z are the standard Pauli operators. In particular, observe that $X_j^2 = Z_j^2 = \mathbf{1}$, $X_j Z_j = -Z_j X_j$.

The above construction satisfies that if $|u_1\rangle, |v_1\rangle, |u_2\rangle, |v_2\rangle$ are pairwise orthogonal, then $[X_1, X_2] = [X_1, Z_2] = [Z_1, X_2] = [Z_1, Z_2] = 0$. The reason we use two vectors to define each X_j, Z_j (instead of just taking $X = a_u + a_u^\dagger$, $Z = 2a_u a_u^\dagger - \mathbf{1}$) is to obtain the above commutation relationships. Since X_1, Z_1 are each quadratic in $a_{u_1}, a_{u_1}^\dagger, a_{v_1}, a_{v_1}^\dagger$, terms involving only $a_{u_2}, a_{u_2}^\dagger, a_{v_2}, a_{v_2}^\dagger$ commute past them.

Next, for *nearly* orthogonal planes we will show that the commutator norm $\|[S_i, T_j]\| = O(\epsilon)$, for $i \neq j$ and $S, T \in \{X, Z\}$.

If $|u\rangle, |v\rangle$ are orthonormal, and $|t\rangle = \epsilon|u\rangle + \sqrt{1-\epsilon^2}|v\rangle$, then

$$a_t = \epsilon a_u + \sqrt{1-\epsilon^2} a_v = \begin{pmatrix} 0 & \sqrt{1-\epsilon^2} & \epsilon & 0 \\ 0 & 0 & 0 & \epsilon \\ 0 & 0 & 0 & -\sqrt{1-\epsilon^2} \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

satisfies $\{a_t, a_u\} = 0$, $\{a_t, a_u^\dagger\} = \epsilon \mathbf{1}$. In general,

$$\begin{aligned} \{a_t, a_u\} &= 0 \\ \{a_t, a_u^\dagger\} &= \langle u|t\rangle \mathbf{1} \ . \end{aligned}$$

It follows that if $|\langle u_1|u_2\rangle|, |\langle u_1|v_2\rangle|, |\langle v_1|u_2\rangle|, |\langle v_1|v_2\rangle| \leq \epsilon$, then $\|[S_1, T_2]\| = O(\epsilon)$ for $S, T \in \{X, Z\}$. Indeed,

$$\begin{aligned} X_1 a_{u_2} &= (-a_{u_1} + a_{u_1}^\dagger)(a_{v_1} + a_{v_1}^\dagger) a_{u_2} \\ &= -(-a_{u_1} + a_{u_1}^\dagger)\big[a_{u_2}(a_{v_1} + a_{v_1}^\dagger) - \langle u_2|v_1\rangle \mathbf{1}\big] \\ &= \big[a_{u_2}(-a_{u_1} + a_{u_1}^\dagger) - \langle u_2|u_1\rangle \mathbf{1}\big](a_{v_1} + a_{v_1}^\dagger) + \langle u_2|v_1\rangle(-a_{u_1} + a_{u_1}^\dagger) \\ &= a_{u_2} X_1 - \langle u_2|u_1\rangle(a_{v_1} + a_{v_1}^\dagger) + \langle u_2|v_1\rangle(-a_{u_1} + a_{u_1}^\dagger) \ , \end{aligned}$$

implying $\|[X_1, a_{u_2}]\| \leq |\langle u_2|u_1\rangle| + |\langle u_2|v_1\rangle| \leq 2\epsilon$. Similarly,

$$\begin{aligned} Z_1 a_{u_2} &= (2 a_{u_1} a_{u_1}^\dagger - \mathbf{1}) a_{u_2} \\ &= 2 a_{u_1}(\langle u_1|u_2\rangle \mathbf{1} - a_{u_2} a_{u_1}^\dagger) - a_{u_2} \\ &= a_{u_2} Z_1 + 2|\langle u_1|u_2\rangle| a_{u_1} \ , \end{aligned}$$

implying $\|[Z_1, a_{u_2}]\| \leq 2|\langle u_1|u_2\rangle| \leq 2\epsilon$. Thus $\|[S_1, T_2]\| \leq c\,\epsilon$ for a fairly small constant c. ◀

Quantum Recommendation System

Iordanis Kerenidis[*1] **and Anupam Prakash**[†2]

1 **CNRS, IRIF, Université Paris Diderot, Paris, France and Centre for Quantum Technologies, National University of Singapore, Singapore**
jkeren@liafa.univ-paris-diderot.fr

2 **Centre for Quantum Technologies and School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore**
aprakash@ntu.edu.sg

Abstract

A recommendation system uses the past purchases or ratings of n products by a group of m users, in order to provide personalized recommendations to individual users. The information is modeled as an $m \times n$ preference matrix which is assumed to have a good rank-k approximation, for a small constant k.

In this work, we present a quantum algorithm for recommendation systems that has running time $O(\text{poly}(k)\text{polylog}(mn))$. All known classical algorithms for recommendation systems that work through reconstructing an approximation of the preference matrix run in time polynomial in the matrix dimension. Our algorithm provides good recommendations by sampling efficiently from an approximation of the preference matrix, without reconstructing the entire matrix. For this, we design an efficient quantum procedure to project a given vector onto the row space of a given matrix. This is the first algorithm for recommendation systems that runs in time polylogarithmic in the dimensions of the matrix and provides an example of a quantum machine learning algorithm for a real world application.

1998 ACM Subject Classification G.1.3 Numerical Linear Algebra

Keywords and phrases Recommendation systems, quantum machine learning, singular value estimation, matrix sampling, quantum algorithms

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.49

1 Introduction

A recommendation system uses information about past purchases or ratings of products by a group of users in order to provide personalized recommendations to individual users. More precisely, we assume there are m users, for example clients of an online platform like Amazon or Netflix, each of whom have some inherent preference or utility about n products, for example books, movies etc. The user preferences are modeled by an $m \times n$ matrix P, where the element P_{ij} denotes how much the user i values product j. If the preference matrix P had been known in advance, it would have been easy to make good recommendations to the users by selecting elements of this matrix with high value. However, this matrix is not known a priori. Information about P arrives in an online manner each time a user buys a product, writes a review, or fills out a survey. A recommendation system tries to utilize the

* IK was partially supported by projects ANR RDAM, ERC QCC and EU QAlgo.
† AP was supported by the Singapore National Research Foundation under NRF RF Award No. NRF-NRFF2013-13.


© Iordanis Kerenidis and Anupam Prakash;
licensed under Creative Commons License CC-BY

already known information about all users in order to suggest products to individual users that have high utility for them and can eventually lead to a purchase.

There has been an extensive body of work on recommendation systems, since it is a very interesting theoretical problem and also of great importance to the industry. We cite the works of [5, 18, 8, 4] who studied the problem in a combinatorial or linear algebraic fashion. There has also been a series of works in the machine learning community many of them inspired by a practical challenge by Netflix on real world data [14, 6, 13].

We next discuss the low rank assumption on the preference matrix underlying recommendation systems and the way this assumption is used to perform matrix reconstruction in classical recommendation systems. We then describe the computational model for our quantum recommendation algorithm that is based on matrix sampling and compare it to classical recommendation algorithms based on matrix reconstruction. We provide a high level overview of our algorithm in section 1.1 and then, we compare it with previous work on quantum machine learning in section 1.2.

The low-rank assumption

The underlying assumption in recommendation systems is that one can infer information about a specific user from the information about all other users because, in some sense, the majority of users belong to some well-defined “types". In other words, most people's likes are not unique but fall into one of a small number of categories. Hence, we can aggregate the information of “similar" users to predict which products have high utility for an individual user.

More formally, the assumption in recommendation systems is that the preference matrix P can be well approximated (according to some distance measure) by a low-rank matrix. There are different reasons why this assumption is indeed justified. First, from a philosophical and cognitive science perspective, it is believed that there are few inherent reasons why people buy or not a product: the price, the quality, the popularity, the brand recognition, etc. (see for example [5, 21]). Each user can be thought of as weighing these small number of properties differently but still, his preference for a product can be computed by checking how the product scores in these properties. Such a model produces a matrix P which has a good rank-k approximation, for a small k, which can be thought of as a constant independent of the number of users m or the number of products n. Moreover, a number of theoretical models of users have been proposed in the literature which give rise to a matrix with good low-rank approximation. For example, if one assumes that the users belong to a small number of “types", where a type can be thought of as an archetypical user, and then each individual user belonging to this type is some noisy version of this archetypical user, then the matrix has a good low-rank approximation [8, 18]. In addition, preference matrices that come from real data have been found to have rank asymptotically much smaller than the size of the matrix.

For these reasons, the assumption that the matrix P has a good low-rank approximation has been widely used in the literature. In fact, if we examine this assumption more carefully, we find that in order to justify that the recommendation system provides high-value recommendations, we assume that users “belong" to a small number of user types and also that they agree with these types on the high-value elements. For contradiction, imagine that there are k types of users, where each type has very few high-value elements and many small value elements. Then, the users who belong to each type can agree on all the small value elements and have completely different high-value elements. In other words, even though the matrix is low-rank, the recommendations would be of no quality. Hence, the assumption

that has been implicitly made, either by philosophical reasons or by modeling the users, is that there are k types of users and the users of each type "agree" on the high-value elements.

Recommendations by Matrix Reconstruction

One of the most powerful and common ways to provide competitive recommendation systems is through a procedure called matrix reconstruction. In this framework, we assume that there exists a hidden matrix A, in our case the preference matrix, which can be well approximated by a low-rank matrix. The reconstruction algorithm gets as input a number of samples from A, in our case the previous data about the users' preferences, and outputs a rank-k matrix with the guarantee that it is "close" to A according to some measure (for example, the 2- or the Frobenius norm). For example, the reconstruction algorithm can perform a Singular Value Decomposition on the subsample matrix $\widehat{A}$, where $\widehat{A}$ agrees with A on known samples and is 0 on the remaining entries, and output the projection of $\widehat{A}$ onto the space spanned by its top-k singular vectors. The "closeness" property guarantees that the recommendation system will select an element that with high probability corresponds to a high-value element in the matrix A and hence it is a good recommendation ([8, 5]). Another commonly used algorithm for matrix reconstruction is a variant of alternating minimization, this has been successful in practice [14] and has been recently analyzed theoretically [11]. Note that all known algorithms for matrix reconstruction require time polynomial in the matrix dimensions.

An important remark is that matrix reconstruction is a harder task than recommendation systems, in the sense that a good recommendation system only needs to output a high value element of the matrix and not the entire matrix [20, 3]. Nevertheless, classical algorithms perform a reconstruction of the entire matrix as the resources required for finding high value elements are the same as the resources needed for full reconstruction.

Computational resources and performance

In order to make a precise comparison between classical recommendation systems and our proposed system, we discuss more explicitly the computational resources in recommendation systems. We are interested in systems that arise in the real world, for example on Amazon or Netflix, where the number of users can be about 100 million and the products around one million. For such large systems, storing the entire preference matrix or doing heavy computations every time a user comes into the system is prohibitive.

The memory model for an online recommendation system is the following. A data structure is maintained that contains the information that arrives into the system in the form of elements P_{ij} of the preference matrix. We require that the time needed to write the tuple (i, j, P_{ij}) into the memory data structure and to read it out is polylogarithmic in the matrix dimensions. In addition, we require that the total memory required is linear (up to polylogarithmic terms) in the number of entries of the preference matrix that have arrived into the system. For example, one could store the elements (i, j, P_{ij}) in an ordered list.

Most classical recommendation systems use a two stage approach. The first stage involves preprocessing the data stored in memory. For example, a matrix reconstruction algorithm can be performed during the preprocessing stage to produce and store a low-rank approximation of the preference matrix. This computation takes time polynomial in the matrix dimensions, poly(mn), and the stored output is the top-k row singular vectors that need space $O(nk)$. The second stage is an online computation that is performed when a user comes into the system. For example, one can project the row of the subsample matrix that corresponds to

this user onto the already stored top-k row singular vectors of the matrix and output a high value element in time $O(nk)$.

The goal is to minimize the time needed to provide an online recommendation while at the same time keeping the time and the extra memory needed for the preprocessing reasonable. In general, the preprocessing time is polynomial in the dimensions of the preference matrix, i.e. poly(mn), the extra memory is $O(nk)$, while the time for the online recommendation is $O(nk)$. Note that in real world applications, it is prohibitive to have a system where the preprocessing uses memory $O(mn)$, even though with such large memory the online recommendation problem becomes trivial as all the answers can be pre-computed and stored.

A recommendation system performs well, when with high probability and for most users it outputs a recommendation that is good for a user. The performance of our recommendation system is similar to previous classical recommendation systems based on matrix reconstruction and depends on how good the low-rank approximation of the matrix is. Our algorithm works for any matrix, but as in the classical case, it guarantees good recommendations only when the matrix has a good low-rank approximation.

1.1 Our results

In this section, we provide a high level overview of our quantum recommendation algorithm which requires time polylogarithmic in the matrix dimensions and polynomial only in the rank of the matrix, which as we have argued is assumed to be much smaller than the dimension of the matrix. This is the first algorithm for recommendation systems with complexity polylogarithmic in the matrix dimensions.

Our model

First, we describe a simple and general model for online recommendation systems. We start with a hidden preference matrix T, where the element T_{ij} takes values 0 or 1 and indicates whether product j is "good" for user i. Such boolean matrices arise naturally in a "thumbs up / thumbs down" system, where users can declare whether they like or not a certain product. We can also easily construct such matrices from non-boolean preference matrices. For each user, we split the products into two categories, the “good" and the “bad" recommendations, based on the matrix entries. This categorization can be done in different ways and we do not have to impose any constraints. For example, good recommendations can be every product with value higher than a threshold or the 100 products with the highest values etc.

Our assumption is that the matrix T has a good low-rank approximation. The reasons that justify this assumption are the ones used already in the literature. As before, we believe that there is a small number of user types, and within each type the users “agree" on the high-value elements. This modelling of the users gives rise to a matrix T with a good low-rank approximation. Once we have defined the matrix T, then any algorithm that reconstructs a matrix close to T, will provide a good recommendation, since T is the indicator matrix of good recommendations.

Recommendations by Matrix Sampling

The low-rank approximation of the matrix T can be computed as follows: first, define the matrix $\widehat{T}$, where with some probability each element of $\widehat{T}$ is equal to the corresponding element in T normalized and otherwise it is zero. This matrix, that we call a subsample matrix, corresponds to the information the recommendation system has already gathered about the matrix T. Then, by performing a Singular Value Decomposition and computing

the projection of this matrix to its top-k row singular vectors, we compute a matrix $\widehat{T}_k$ which can be proven to be close to the matrix T, as long as T had a good rank-k approximation.

As we remarked, in principle, we do not need to explicitly compute the entire matrix $\widehat{T}_k$. It is sufficient to be able to sample from the matrix $\widehat{T}_k$ which is close to T. Since T is a 0-1 matrix, sampling from $\widehat{T}_k$ means finding with high probability a 1-element in T. By the fact that T indicates the good recommendations, our algorithm will output a good recommendation with high probability. Hence, we reduce the question of providing good recommendations to being able to sample from the matrix $\widehat{T}_k$. In fact, since we want to be able to recommend products to any specific user i, we need to be able, given an index i, to sample from the i-th row of the matrix $\widehat{T}_k$, denoted by $(\widehat{T}_k)_i$, i.e. output an element $(\widehat{T}_k)_{ij}$ with probability $|(\widehat{T}_k)_{ij}|^2/\|(\widehat{T}_k)_i\|^2$. Note that the row $(\widehat{T}_k)_i$ is the projection of the row $\widehat{T}_i$ onto the top-k row singular vectors of $\widehat{T}$.

An efficient quantum algorithm for Matrix Sampling

Here is where quantum computing becomes useful: we design a quantum procedure that samples from the row $(\widehat{T}_k)_i$ in time polylog(mn). Note that the quantum algorithm does not output the row $(\widehat{T}_k)_i$, which by itself would take time linear in the dimension n, but only samples from this row. But this is exactly what is needed for recommendation systems: Sample a high-value element of the row, rather than explicitly output the entire row. More precisely, we describe an efficient quantum procedure that takes as input a vector, a matrix, and a threshold parameter and generates the quantum state corresponding to the projection of the vector onto the space spanned by the row singular vectors of the matrix whose corresponding singular value is greater than the threshold. From the outcome of this procedure it is clear how to sample a product by just measuring the quantum state in the computational basis.

1.2 Comparisons with related work

The development of quantum algorithms for linear algebra was initiated by the breakthrough algorithm of Harrow, Hassidim, Lloyd [10]. The HHL algorithm takes as input a sparse (the number of non zero entries in each row of the matrix is polylogarithmic) and well-conditioned system of linear equations and in time polylogarithmic in the dimension of the system outputs a quantum state which corresponds to the classical solution of the system. Note that this algorithm does not explicitly output the classical solution, nevertheless, the quantum state enables one to sample from the solution vector. This is a very powerful algorithm and has been very influential in recent times, where several works [16, 15, 17] obtained quantum algorithms for machine learning problems based on similar assumptions. However, when looking at these applications, one needs to be extremely careful about two things: first, the assumptions that one needs to make on the input in order to achieve efficient running time, since, for example, the running time of the HHL algorithm is polylogarithmic only when the matrix is well conditioned (i.e. the minimum singular value is at least inverse polynomially big) and sparse; and second, whether the quantum algorithm solves the original classical problem or a weaker variant to account for the fact that the classical solution is not given explicitly but is encoded in a quantum state [1, 17]. In addition, we mention a recent but orthogonal proposal to use techniques inspired by the structure of quantum theory for classical recommender systems [22].

Let us be more explicit about our algorithm's assumptions. We assume the data is stored in a classical data structure which enables the quantum algorithm to efficiently create

superpositions of rows of the subsample matrix. The HHL algorithm also needs to be able to efficiently construct quantum states from classical vectors given as inputs. In the Appendix, we describe a classical data structure for storing the matrix $\widehat{T}$. The data structure maintains some extra information about the matrix entries, so that, the total memory needed is linear (up to polylogarithmic terms) in the number of entries in the subsample matrix, the data entry time remains polylogarithmic in the matrix dimensions, and an algorithm with quantum access to the data structure can create the necessary superpositions in polylogarithmic time. Note also, that even in the case the data has been stored as a normal array or list, we can preprocess it in linear time to construct our needed data structure. Thus, our quantum algorithm works under the same memory model as any other quantum query algorithm (e.g. Grover's algorithm): it assumes that there exists a classical data structure to which we can make quantum queries. Overall, our system retains the necessary properties for the data entry and retrieval stage. Moreover, the classical complexity of matrix reconstruction does not change given the new data structure.

Importantly, in our system, we do not perform any preprocessing nor do we need any extra memory. Our recommendation algorithm just performs an online computation that requires time $\text{poly}(k)\text{polylog}(mn)$. This can be viewed as exponentially smaller than the classical time if the rank k is a small constant and the matrix dimensions are of the same order. Unlike the HHL algorithm, our running time does not depend on the sparsity of the input matrix nor on its condition number, i.e. its smallest singular value. In other words, we do not make any additional assumptions about the classical data beyond the low rank approximation assumptions made by classical recommendation systems.

It is also crucial to note that we have not changed what one needs to output, as was the case for the HHL algorithm and its applications, where instead of explicitly outputting a classical solution, they construct a quantum state that corresponds to this solution. We have instead described a real world application, where the ability to sample from the solution is precisely what is needed. Our work also suggests a new avenue for classical recommendation systems, which we state below as an open problem.

▶ **Open Problem 1.** *Find a classical recommendation algorithm using matrix sampling that requires time $O(poly(k)polylog(mn))$ or prove a lower bound that rules out the existence of such an algorithm.*

The rest of this paper is organized as follows. We introduce some preliminaries in section 2. In sections 3 and 4 we show that sampling from an approximate reconstruction of the matrix T suffices to provide good recommendations and that if the sub-samples $\widehat{T}$ are uniformly distributed then projecting onto the top k singular vectors of $\widehat{T}$ is an approximate reconstruction for T. In section 5 we describe an efficient quantum algorithm for projecting a vector onto the space of singular vectors of $\widehat{T}$ whose corresponding singular values are greater than a threshold. In section 6 we combine these components to obtain a quantum recommendation algorithm and analyze its performance and running time.

2 Preliminaries

2.1 Linear algebra

The set $\{1, 2, \cdots, n\}$ is denoted by $[n]$, the standard basis vectors in $\mathbb{R}^n$ are denoted by $e_i, i \in [n]$. For any matrix $A \in \mathbb{R}^{m\times n}$, the Frobenius norm is defined as $\|A\|_F^2 = \sum_{ij} A_{ij}^2 = \sum_i \sigma_i^2$, where σ_i are the singular values. We also say that we sample from the matrix A

when we pick an element (i,j) with probability $|A_{ij}|^2/\|A\|_F^2$, and write $(i,j) \sim A$. For a vector $x \in \mathbb{R}^n$ we denote the norm $\|x\|^2 = \sum_i x_i^2$.

The matrix A is unitary if $AA^* = A^*A = I$, the eigenvalues of a unitary matrix have unit norm. A matrix $P \in \mathbb{R}^{n\times n}$ is a projector if $P^2 = P$. If A is a matrix with orthonormal columns, then AA^t is the projector onto the column space of A.

Singular value decomposition The singular value decomposition of $A \in \mathbb{R}^{m\times n}$ is a decomposition of the form $A = U\Sigma V^t$ where $U \in \mathbb{R}^{m\times m}, V \in \mathbb{R}^{n\times n}$ are unitary and $\Sigma \in \mathbb{R}m \times n$ is a diagonal matrix with positive entries. The SVD can be written as $A = \sum_{i\in[r]} \sigma_i u_i v_i^t$ where r is the rank of A. The column and the row singular vectors u_i and v_i are the columns of U and V respectively. The Moore Penrose pseudo-inverse is defined as $A^+ = V\Sigma^+U^t$, where $A^+ = \sum_{i\in[r]} \frac{1}{\sigma_i} v_i u_i^t$. It follows that AA^+ is the projection onto the column space $Col(A)$ while A^+A is the projection onto the row space $Row(A)$. The truncation of A to the space of the singular vectors that correspond to the k largest singular values is denoted by A_k, that is $A_k = \sum_{i\in[k]} \sigma_i u_i v_i^t$. We denote by $A_{\geq\sigma}$ the projection of the matrix A onto the space spanned by the singular vectors whose corresponding singular value is bigger than σ, that is $A_{\geq\sigma} = \sum_{i:\sigma_i\geq\sigma} \sigma_i u_i v_i^t$.

2.2 Quantum information

We use the standard bra-ket notation to denote quantum states. We use the following encoding for representing n dimensional vectors by quantum states,

▶ **Definition 2.** *The vector state $|x\rangle$ for $x \in \mathbb{R}^n$ is defined as $\frac{1}{\|x\|}\sum_{i\in[n]} x_i |i\rangle$.*

In case $x \in \mathbb{R}^{mn}$, we can either see it as a vector in this space or as a matrix with dimensions $m \times n$ and then we can equivalently write $\frac{1}{\|x\|}\sum_{i\in[m],j\in[n]} x_{ij} |i,j\rangle$.

A quantum measurement ($POVM$) is a collection of positive operators $M_a \succeq 0$ such that $\sum_a M_a = I_n$. The probability of obtaining outcome a when state $|\phi\rangle$ is measured is $Tr(\langle\phi|M_a\phi\rangle)$. If $|x\rangle$ is measured in the standard basis, then outcome i is observed with probability $x_i^2/\|x\|^2$.

We also use a well-known quantum algorithm called phase estimation. The phase estimation algorithm estimates the eigenvalues of a unitary operator U with additive error ϵ in time $O(T(U)\log n/\epsilon)$ if $T(U)$ is the time required to implement the unitary U.

▶ **Theorem 3.** Phase estimation [12]*: Let U be a unitary operator, with eigenvectors $|v_j\rangle$ and eigenvalues $e^{\iota\theta_j}$ for $\theta_j \in [-\pi,\pi]$, i.e. we have $U|v_j\rangle = e^{\iota\theta_j}|v_j\rangle$ for $j \in [n]$. For a precision parameter $\epsilon > 0$, there exists a quantum algorithm that runs in time $O(T(U)\log n/\epsilon)$ and with probability $1-1/\mathrm{poly}(n)$ maps a state $|\phi\rangle = \sum_{j\in[n]} \alpha_j |v_j\rangle$ to the state $\sum_{j\in[n]} \alpha_j |v_j\rangle|\overline{\theta_j}\rangle$ such that $\overline{\theta_j} \in \theta_j \pm \epsilon$ for all $j \in [n]$.*

Note that we use ι to denote the imaginary unit i to avoid confusion with summation indices. The analysis of phase estimation shows that the algorithm outputs a discrete valued estimate for each eigenvalue that is within additive error ϵ with probability at least 0.8, the probability is boosted to $1 - 1/\mathrm{poly}(n)$ by repeating $O(\log n)$ times and choosing the most frequent estimate.

3 A model for recommendation systems

3.1 The preference matrix

We define a simple and general model for recommendation systems. We define a *preference matrix* T of size $m \times n$, where every row corresponds to a user, every column to a product, and the element T_{ij} is 0 or 1 and denotes whether product j is a good recommendation for user i or not.

▶ **Definition 4.** *A product j is a good recommendation for user i iff $T_{ij} = 1$, otherwise it is bad. We also write it as the pair (i, j) is a good or bad recommendation.*

Such matrices arise in systems where the information the users enter is binary, for example in a "thumbs up / thumbs down" system. We can also construct such matrices from more general preference matrices where the users use a star system to grade the products. One can imagine, for example, that the good recommendations could be the products for which the user has a preference higher than a threshold, or the hundred products with highest preference etc.

3.2 Sampling an approximation of the preference matrix

Note that sampling from the preference matrix T would always yield a good recommendation, since the products that correspond to bad recommendations have probability 0. This remains true even when we want to sample from a specific row of the matrix in order to provide a recommendation to a specific user. Our goal now is to show that sampling from a matrix that is close to the matrix T under the Frobenius norm yields good recommendations with high probability for most users.

▶ **Lemma 5.** *Let $\widetilde{T}$ be an approximation of the matrix T such that $\|T - \widetilde{T}\|_F \leq \epsilon \|T\|_F$. Then, the probability a sample according to $\widetilde{T}$ is a bad recommendation is*

$$\Pr_{(i,j)\sim\widetilde{T}}[(i,j) \text{ bad}] \leq \left(\frac{\epsilon}{1-\epsilon}\right)^2$$

Proof. By the theorem's assumption and triangle inequality, we have

$$(1+\epsilon)\|T\|_F \geq \|\widetilde{T}\|_F \geq (1-\epsilon)\|T\|_F.$$

We can rewrite the approximation guarantee as

$$\epsilon^2 \|T\|_F^2 \geq \|T - \widetilde{T}\|_F^2 = \sum_{(i,j):good} (1 - \widetilde{T}_{ij})^2 + \sum_{(i,j):bad} \widetilde{T}_{ij}^2 \geq \sum_{(i,j):bad} \widetilde{T}_{ij}^2 \tag{1}$$

The probability that sampling from $\widetilde{T}$ provides a bad recommendation is

$$\Pr[(i,j) \text{ bad}] = \frac{\sum_{(i,j):bad} \widetilde{T}_{ij}^2}{\|\widetilde{T}\|_F^2} \leq \frac{\sum_{(i,j):bad} \widetilde{T}_{ij}^2}{(1-\epsilon)^2 \|T\|_F^2} \leq \left(\frac{\epsilon}{1-\epsilon}\right)^2. \tag{2}$$

◀

The above can be rewritten as follows denoting the i-th row of T by T_i,

$$\Pr[(i,j) \text{ bad}] = \frac{\sum_{(i,j):bad} \widetilde{T}_{ij}^2}{\|\widetilde{T}\|_F^2} = \sum_{i\in[m]} \frac{\|\widetilde{T}_i\|_F^2}{\|\widetilde{T}\|_F^2} \cdot \frac{\sum_{j:(i,j)bad} \widetilde{T}_{ij}^2}{\|\widetilde{T}_i\|_F^2} \leq \left(\frac{\epsilon}{1-\epsilon}\right)^2. \tag{3}$$

We can see that the above lemma provides the guarantee that the probability of a bad recommendation for an average user is small, where the average is weighted according to the weight of each row. In other words, if we care more about users that have many products they like and less for users that like almost nothing, then the sampling already guarantees good performance.

While this might be sufficient in some scenarios, it would be nice to also have a guarantee that the recommendation is good for most users, where now every user has the same importance. Note that only with the closeness guarantee in the Frobenius norm, this may not be true, since imagine the case where almost all rows of the matrix T have extremely few 1s and a few rows have almost all 1s. In this case, it might be that the approximation matrix is close to the preference matrix according to the Frobenius norm, nevertheless the recommendation system provides good recommendations only for the very heavy users and bad ones for almost everyone else.

Hence, if we would like to show that the recommendation system provides good recommendations for most users, then we need to assume that most users are "typical", meaning that the number of products that are good recommendations for them is close to the average. We cannot expect to provide good recommendations for example to users that like almost nothing. One way to enforce this property is, for example, to define good recommendations for each user as the 100 top products, irrespective of how high their utilities are or whether there are even more good products for some users. In what follows we prove our results in most generality, where we introduce parameters for how many users are typical and how far from the average the number of good recommendations of a typical user can be.

▶ **Theorem 6.** *Let T be an $m \times n$ matrix. Let S be a subset of rows of size $|S| \geq (1-\zeta)m$ (for $\zeta > 0$) such that for all $i \in S$,*

$$\frac{1}{1+\gamma}\frac{\|T\|_F^2}{m} \leq \|T_i\|^2 \leq (1+\gamma)\frac{\|T\|_F^2}{m} \tag{4}$$

for some $\gamma > 0$. Let $\widetilde{T}$ be an approximation of the matrix T such that $\|T-\widetilde{T}\|_F \leq \epsilon\|T\|_F$. Then, there exists a subset $S' \subseteq S$ of size at least $(1-\delta-\zeta)m$ (for $\delta > 0$), such that on average over the users in S', the probability that a sample from the row $\widetilde{T}_i$ is a bad recommendation is

$$\Pr_{i\sim\mathcal{U}_{S'}, j\sim\widetilde{T}_i}[(i,j) \text{ bad}] \leq \frac{\left(\frac{\epsilon(1+\epsilon)}{1-\epsilon}\right)^2}{\left(1/\sqrt{1+\gamma} - \epsilon/\sqrt{\delta}\right)^2 (1-\delta-\zeta)}.$$

Proof. We first use the guarantee that the matrices T and $\widetilde{T}$ are close in the Frobenius norm to conclude that there exist at least $(1-\delta)m$ users for which

$$\|T_i - \widetilde{T}_i\|^2 \leq \frac{\epsilon^2\|T\|_F^2}{\delta m}. \tag{5}$$

If not, summing the error of the strictly more than δm users for which equation 5 is false we get the following contradiction,

$$\|T-\widetilde{T}\|_F^2 > \delta m \frac{\epsilon^2\|T\|_F^2}{\delta m} > \epsilon^2\|T\|_F^2.$$

Then, at least $(1-\delta-\zeta)m$ users both satisfy equation 5 and belong to the set S. Denote this set by S'. Using equations (4) and (5) and the triangle inequality $\|\widetilde{T}_i\| \geq \|T_i\| - \|T_i - \widetilde{T}_i\|$, we have that for all users in S'

$$\|\widetilde{T}_i\|_F^2 \geq \frac{\|T\|_F^2}{m}\left(\frac{1}{\sqrt{1+\gamma}} - \frac{\epsilon}{\sqrt{\delta}}\right)^2 \geq \frac{\|\widetilde{T}\|_F^2}{(1+\epsilon)^2 m}\left(\frac{1}{\sqrt{1+\gamma}} - \frac{\epsilon}{\sqrt{\delta}}\right)^2. \tag{6}$$

We now use equations (3) and (6) and have

$$\left(\frac{\epsilon}{1-\epsilon}\right)^2 \geq \sum_{i\in[m]} \frac{\|\widetilde{T}_i\|_F^2}{\|\widetilde{T}\|_F^2} \cdot \frac{\sum_{j:(i,j)bad} \widetilde{T}_{ij}^2}{\|\widetilde{T}_i\|_F^2} \geq \frac{\left(1/\sqrt{1+\gamma} - \epsilon/\sqrt{\delta}\right)^2}{(1+\epsilon)^2 m} \sum_{i\in S'} \frac{\sum_{j:(i,j)bad} \widetilde{T}_{ij}^2}{\|\widetilde{T}_i\|_F^2}. \quad (7)$$

We are now ready to conclude that,

$$\Pr_{i\sim\mathcal{U}_{S'}, j\sim\widetilde{T}_i}[(i,j) \text{ bad}] = \frac{1}{|S'|} \sum_{i\in S'} \frac{\sum_{j:(i,j)bad} \widetilde{T}_{ij}^2}{\|\widetilde{T}_i\|_F^2} \leq \frac{\left(\frac{\epsilon(1+\epsilon)}{1-\epsilon}\right)^2}{\left(1/\sqrt{1+\gamma} - \epsilon/\sqrt{\delta}\right)^2 (1-\delta-\zeta)}.$$

◀

We note that by taking reasonable values for the parameters, the error does not increase much from the original error. For example, if we assume that 90% of the users have preferences between 1/1.1 and 1.1 times the average, then the error over the typical users has increased by at most a factor of 1.5. Note also that we can easily make the quality of the recommendation system even better if we are willing to recommend a small number of products, instead of just one, and are satisfied if at least one of them is a good recommendation. This is in fact what happens in practical systems.

4 Matrix Sampling

We showed in the previous section that providing good recommendations reduces to being able to sample from a matrix $\widetilde{T}$ which is a good approximation to the recommendation matrix T in the Frobenius norm. We will now define the approximation matrix $\widetilde{T}$, by extending known matrix reconstruction techniques. The reconstruction algorithms provides good guarantees under the assumption that the recommendation matrix T has a good k-rank approximation for a small k, i.e. $\|T - T_k\|_F \leq \epsilon\|T\|_F$ (for some small constant $\epsilon \geq 0$).

Let us now briefly describe the matrix reconstruction algorithms. In general, the input to the reconstruction algorithm is a subsample of some matrix A. There are quite a few different ways of subsampling a matrix, for example, sampling each element of the matrix with some probability or sampling rows and/or columns of the matrix according to some distribution. We present here in more detail the first case as is described in the work of Achlioptas and McSherry [2]. Each element of the matrix A that has size $m \times n$ is sampled with probability p and rescaled so as to obtain the random matrix $\widehat{A}$ where each element is equal to $\widehat{A}_{ij} = A_{ij}/p$ with probability p and 0 otherwise. Note that $E[\widehat{A}] = A$ and that it is assumed that k and $\|A\|_F$ are known.

The reconstruction algorithm computes the projection of the input matrix $\widehat{A}$ onto its k-top singular vectors; we denote the projection by $\widehat{A}_k$. The analysis of the algorithm shows that the approximation error $\|A - \widehat{A}_k\|$ is not much bigger than $\|A - A_k\|$. Projecting onto the top k singular vectors of the subsampled matrix $\widehat{A}$ thus suffices to reconstruct a matrix approximating A.

The intuition for the analysis is that $\widehat{A}$ is a matrix whose entries are independent random variables, thus with high probability the top k spectrum of $\widehat{A}$ will be close to the one of its expectation matrix $E[\widehat{A}] = A$. This intuition was proven in [2].

▶ **Theorem 7.** *[2] Let $A \in \mathbb{R}^{m\times n}$ be a matrix and $b = \max_{ij} A_{ij}$. Define the matrix $\widehat{A}$ to be a random matrix obtained by subsampling with probability $p = 16nb^2/(\eta\|A\|_F)^2$ (for $\eta > 0$)*

and rescaling, that is $\widehat{A}_{ij} = A_{ij}/p$ *with probability* p *and* 0 *otherwise. With probability at least* $1 - exp(-19(\log n)^4)$ *we have for any* k

$$\|A - \widehat{A}_k\|_F \leq \|A - A_k\|_F + 3\sqrt{\eta}k^{1/4}\|A\|_F. \tag{8}$$

Here, we will need to extend this result in order to be able to use it together with our quantum procedure. First, we will consider the matrix which is not the projection on the k-top singular vectors, but the projection on the singular vectors whose corresponding singular values are larger than a threshold. For any matrix A and any $\sigma \geq 0$, we denote by $A_{\geq\sigma}$ the projection of the matrix A onto the space spanned by the singular vectors whose corresponding singular value is bigger than σ. Intuitively, since the spectrum of the matrix is highly concentrated on the top k singular vectors, then the corresponding singular values should be of order $O(\frac{\|A\|_F}{\sqrt{k}})$.

Note that we do not use anything about how the matrix $\widehat{A}$ was generated, only that it satisfies equation 8. Hence our results hold for other matrix reconstruction algorithms as well, as long as we have a similar guarantee in the Frobenius norm.

▶ **Theorem 8.** *Let* $A \in \mathbb{R}^{m\times n}$ *be a matrix such that* $\max_{ij} A_{ij} = 1$. *Define the matrix* $\widehat{A}$ *to be a random matrix obtained by subsampling with probability* $p = 16n/\eta^2(\|A\|_F)^2$ *(for* $\eta > 0$*) and rescaling, that is* $\widehat{A}_{ij} = A_{ij}/p$ *with probability* p *and* 0 *otherwise. Let* $\mu > 0$ *a threshold parameter and denote* $\sigma = \sqrt{\frac{\mu}{k}}||\widehat{A}||_F$. *With probability at least* $1 - exp(-19(\log n)^4)$ *we have*

$$\|A - \widehat{A}_{\geq\sigma}\|_F \leq \|A - A_k\|_F + (3\sqrt{\eta}k^{1/4}\mu^{-1/4} + \sqrt{\mu/p})\|A\|_F. \tag{9}$$

If $\|A - A_k\|_F \leq \epsilon\|A\|_F$ *for some* $\epsilon > 0$ *and* $\|A\|_F \geq \frac{36\sqrt{2}(nk)^{1/2}}{\epsilon^3}$ *then we can choose* η, μ *such that* $\|A - \widehat{A}_{\geq\sigma}\|_F \leq 3\epsilon\|A\|_F$.

Proof. Let σ_i denote the singular values of $\widehat{A}$. Let ℓ the largest integer for which $\sigma_\ell \geq \sqrt{\frac{\mu}{k}}||\widehat{A}||_F$. Note that $\ell \leq \frac{k}{\mu}$. Then, by theorem 7, we have

$$\|A - \widehat{A}_{\geq\sigma}\|_F = \|A - \widehat{A}_\ell\|_F \leq \|A - A_\ell\|_F + 3\sqrt{\eta}\ell^{1/4}\|A\|_F.$$

Define the random variable $X = \sum_{i,j}\widehat{A}_{ij}^2$ so that $X = \|\widehat{A}\|_F^2$ and $E[X] = \|A\|_F^2/p$. The random variables $\widehat{A}_{ij}$ are independent, using the Chernoff bounds we have $\Pr[\|\widehat{A}\|_F^2 > (1+\beta)\|A\|_F^2/p] \leq e^{-\beta^2\|A\|_F^2/3p}$ for $\beta \in [0,1]$. The probability that $\|\widehat{A}\|_F^2 > 2\|A\|_F^2/p$ is exponentially small.

We distinguish two cases.

If $\ell \geq k$, then $\|A - A_\ell\|_F \leq \|A - A_k\|_F$, since A_ℓ contains more of the singular vectors of A.

If $k > \ell$, then $\|A - A_\ell\|_F \leq \|A - A_k\|_F + \|A_k - A_\ell\|_F$, which dominates the two cases. For the second term we have $\|A_k - A_\ell\|_F^2 = \sum_{i=\ell+1}^{k}\sigma_i^2 \leq k\frac{\mu}{k}\|\widehat{A}\|_F^2 \leq \frac{2\mu}{p}\|A\|_F^2$. Hence,

$$\|A - \widehat{A}_{\geq\sigma}\|_F \leq \|A - A_k\|_F + (3\sqrt{\eta}k^{1/4}\mu^{-1/4} + \sqrt{2\mu/p})\|A\|_F.$$

If $\|A - A_k\|_F \leq \epsilon\|A\|_F$, for some $\epsilon \geq 0$ then we choose $\mu = \epsilon^2 p/2$ and we can select any $\eta \leq \frac{2n^{1/4}\epsilon^{3/2}}{3(2k)^{1/4}\|A\|_F^{1/2}}$ so that $3\sqrt{\eta}k^{1/4}\mu^{-1/4} \leq \epsilon$ and the overall error $\|A - \widehat{A}_{\geq\sigma}\|_F \leq 3\epsilon\|A\|_F$. Indeed,

$$3\sqrt{\eta}k^{1/4}\mu^{-1/4} = \frac{3\eta^{1/2}(2k)^{1/4}}{\epsilon^{1/2}p^{1/4}} = \frac{3\eta\|A\|_F^{1/2}(2k)^{1/4}}{2\epsilon^{1/2}n^{1/4}} \leq \epsilon$$

Note that for this choice of μ and η, the sampling probability must be at least $p \geq \frac{36\sqrt{2}(nk)^{1/2}}{\|A\|_F\epsilon^3}$, the assumption in the theorem statement ensures that $p \leq 1$. ◀

Our quantum procedure will almost produce this projection. In fact, we will need to consider a family of matrices which denote the projection of the matrix A onto the space spanned by the union of the singular vectors whose corresponding singular value is bigger than σ and also some subset of singular vectors whose corresponding singular value is in the interval $[(1-\kappa)\sigma, \sigma)$. Think of κ as a constant, for example $1/3$. This subset could be empty, all such singular vectors, or any in-between subset. We denote by $A_{\geq\sigma,\kappa}$ any matrix in this family.

The final theorem we will need is the following

▶ **Theorem 9.** *Let $A \in \mathbb{R}^{m\times n}$ be a matrix and $\max_{ij} A_{ij} = 1$. Define the matrix $\widehat{A}$ to be a random matrix obtained by subsampling with probability $p = 16n/(\eta\|A\|_F)^2$ and rescaling, that is $\widehat{A}_{ij} = A_{ij}/p$ with probability p and 0 otherwise. Let $\mu > 0$ a threshold parameter and denote $\sigma = \sqrt{\frac{\mu}{k}}||\widehat{A}||_F$. Let $\kappa > 0$ a precision parameter. With probability at least $1 - exp(-19(\log n)^4)$,*

$$\|A - \widehat{A}_{\geq\sigma,\kappa}\|_F \leq 3\|A - A_k\|_F + \left(3\sqrt{\eta}k^{1/4}\mu^{-1/4}(2 + (1-\kappa)^{-1/2}) + (3-\kappa)\sqrt{2\mu/p}\right)\|A\|_F. \tag{10}$$

If $\|A - A_k\|_F \leq \epsilon\|A\|_F$ for some $\epsilon > 0$ and $\|A\|_F \geq \frac{36\sqrt{2}(nk)^{1/2}}{\epsilon^3}$ then we can choose η, μ such that $\|A - \widehat{A}_{\geq\sigma,\kappa}\|_F \leq 9\epsilon\|A\|_F$.

Proof. We have

$$\begin{aligned}
\|A - \widehat{A}_{\geq\sigma,\kappa}\|_F &\leq \|A - \widehat{A}_{\geq\sigma}\|_F + \|\widehat{A}_{\geq\sigma} - \widehat{A}_{\geq\sigma,\kappa}\|_F \\
&\leq \|A - \widehat{A}_{\geq\sigma}\|_F + \|\widehat{A}_{\geq\sigma} - \widehat{A}_{\geq(1-\kappa)\sigma}\|_F \\
&\leq \|A - \widehat{A}_{\geq\sigma}\|_F + \|A - \widehat{A}_{\geq\sigma}\|_F + \|A - \widehat{A}_{\geq(1-\kappa)\sigma}\|_F \\
&\leq 2\|A - \widehat{A}_{\geq\sigma}\|_F + \|A - \widehat{A}_{\geq(1-\kappa)\sigma}\|_F.
\end{aligned}$$

We use Theorem 8 to bound the first term as

$$\|A - \widehat{A}_{\geq\sigma}\|_F \leq \|A - A_k\|_F + (3\sqrt{\eta}k^{1/4}\mu^{-1/4} + \sqrt{2\mu/p})\|A\|_F$$

For the second term, we can reapply Theorem 8 where now we need to rename μ as $(1-\kappa)^2\mu$ and have

$$\|A - \widehat{A}_{\geq(1-\kappa)\sigma}\|_F \leq \|A - A_k\|_F + (3\sqrt{\eta}k^{1/4}(1-\kappa)^{-1/2}\mu^{-1/4} + (1-\kappa)\sqrt{2\mu/p})\|A\|_F.$$

Overall we have

$$\|A - \widehat{A}_{\geq\sigma,\kappa}\|_F \leq 3\|A - A_k\|_F + \left(3\sqrt{\eta}k^{1/4}\mu^{-1/4}(2 + (1-\kappa)^{-1/2}) + (3-\kappa)\sqrt{2\mu/p}\right)\|A\|_F.$$

Let $\|A - A_k\|_F \leq \epsilon\|A\|_F$, for some $\epsilon \geq 0$. We choose $\kappa = 1/3$, $\mu = \epsilon^2 p/2$ and we can select any $\eta \leq \frac{2n^{1/4}\epsilon^{3/2}}{3(2k)^{1/4}\|A\|_F^{1/2}}$ to have

$$\|A - \widehat{A}_{\geq\sigma,\kappa}\|_F \leq 3\epsilon\|A\|_F + \left(2\epsilon + \frac{\epsilon}{\sqrt{1-\kappa}} + (3-\kappa)\epsilon\right)\|A\|_F \leq 9\epsilon\|A\|_F. \tag{11}$$

As in theorem 8, the sampling probability must be at least $p \geq \frac{36\sqrt{2}(nk)^{1/2}}{\|A\|_F\epsilon^3}$. ◀

We have shown that the task of providing good recommendations for a user i reduces to being able to sample from the i-th row of the matrix $\widehat{T}_{\geq\sigma,\kappa}$, in other words sample from the projection of the i-th row of $\widehat{T}$ onto the space spanned by all row singular vectors with singular values higher than σ and possibly some more row singular vectors with singular values in the interval $[(1-\kappa)\sigma, \sigma)$.

In the following section, we show a quantum procedure, such that given a vector (e.g. the i-th row of $\widehat{T}$), a matrix (e.g. the matrix $\widehat{T}$), and parameters σ and κ, outputs the quantum state $|(\widehat{T}_{\geq\sigma,\kappa})_i\rangle$, which allows one to sample from this row by measuring in the computational basis. The algorithm runs in time polylogarihmic in the matrix dimensions and polynomial in k, since it depends inverse polynomially in σ, which in our case is inverse polynomial in k.

5 Quantum projections in polylogarithmic time

The main quantum primitive required for the recommendation system is a quantum projection algorithm that runs in time polylogarithmic in the matrix dimensions.

5.1 The data structure

The input to the quantum procedure is a vector $x \in \mathbb{R}^n$ and a matrix $A \in \mathbb{R}^{m\times n}$. We assume that the input is stored in a classical data structure such that an algorithm that has quantum access to the data structure can create the quantum state $|x\rangle$ corresponding to the vector x and the quantum states $|A_i\rangle$ corresponding to each row A_i of the matrix A, in time $\text{polylog}(mn)$.

It is in fact possible to design a data structure for a matrix A that supports the efficient construction of the quantum states $|A_i\rangle$. Moreover, we can ensure that the size of the data structure is optimal (up to polylogarithmic factors), and the data entry time, i.e. the time to store a new entry (i, j, A_{ij}) that arrives in the system is just $\text{polylog}(mn)$. Note that just writing down the entry takes logarithmic time.

▶ **Theorem 10.** *Let $A \in \mathbb{R}^{m\times n}$ be a matrix. Entries (i, j, A_{ij}) arrive in the system in an arbitrary order and w denotes the number of entries that have already arrived in the system. There exists a data structure to store the entries of A with the following properties:*

- **(i)** *The size of the data structure is $O(w \cdot \log^2(mn))$.*
- **(ii)** *The time to store a new entry (i, j, A_{ij}) is $O(\log^2(mn))$.*
- **(iii)** *A quantum algorithm that has quantum access to the data structure can perform the mapping $\widetilde{U} : |i\rangle|0\rangle \to |i\rangle|A_i\rangle$, for $i \in [m]$, corresponding to the rows of the matrix currently stored in memory and the mapping $\widetilde{V} : |0\rangle|j\rangle \to |\widetilde{A}\rangle|j\rangle$, for $j \in [n]$, where $\widetilde{A} \in \mathbb{R}^m$ has entries $\widetilde{A}_i = \|A_i\|$ in time $\text{polylog}(mn)$.*

The explicit description of the data structure is given in the appendix. Basically, for each row of the matrix, that we view as a vector in $\mathbb{R}^n$, we store an array of $2n$ values as a full binary tree of n leaves. The leaves hold the individual amplitudes of the vector and each internal node holds the sum of the squares of the amplitudes of the leaves rooted on this node. For each entry added to the tree, we need to update $\log(n)$ nodes in the tree. The same data structure can of course be used for the vector x as well. One need not use a fixed array of size $2n$ for this construction, but only ordered lists of size equal to the entries that have already arrived in the system.Alternative solutions for vector state preparation are possible, another solution based on a modified memory is described in [19].

5.2 Quantum Singular Value Estimation

The second tool required for the projection algorithm is an efficient quantum algorithm for singular value estimation. In the singular value estimation problem we are given a matrix A such that the vector states corresponding to its row vectors can be prepared efficiently. Given a state $|x\rangle = \sum_i \alpha_i |v_i\rangle$ for an arbitrary vector $x \in \mathbb{R}^n$ the task is to estimate the singular values corresponding to each singular vector in coherent superposition. Note that we take the basis $\{v_i\}$ to span the entire space by including singular vectors with singular value 0.

▶ **Theorem 11.** *Let $A \in \mathbb{R}^{m\times n}$ be a matrix with singular value decomposition $A = \sum_i \sigma_i u_i v_i^t$ stored in the data structure in theorem 10. Let $\epsilon > 0$ be the precision parameter. There is an algorithm with running time $O(\text{polylog}(mn)/\epsilon)$ that performs the mapping $\sum_i \alpha_i |v_i\rangle \to \sum_i \alpha_i |v_i\rangle |\overline{\sigma_i}\rangle$, where $\overline{\sigma_i} \in \sigma_i \pm \epsilon\|A\|_F$ for all i with probability at least $1 - 1/\text{poly}(n)$.*

Here, we present a quantum singular value estimation algorithm, in the same flavor as the quantum walk based algorithm by Childs [7] for estimating eigenvalues of a matrix, and show that given quantum access to the data structure from theorem 10, our algorithm runs in time $O(\text{polylog}(mn)/\epsilon)$. A different quantum algorithm for singular value estimation can be based on the work of [16] with running time $O(\text{polylog}(mn)/\epsilon^3)$, and for which a coherence analysis was shown in [19].

The idea for our singular value estimation algorithm is to find isometries $P \in \mathbb{R}^{mn\times m}$ and $Q \in \mathbb{R}^{mn\times n}$ that can be efficiently applied, and such that $\frac{A}{\|A\|_F} = P^tQ$. Using P and Q, we define a unitary matrix W acting on $\mathbb{R}^{mn}$, which is also efficiently implementable and such that the row singular vector v_i of A with singular value σ_i is mapped to an eigenvector Qv_i of W with eigenvalue $e^{\iota\theta_i}$ such that $\cos(\theta_i/2) = \sigma_i/\|A\|_F$ (note that $\cos(\theta_i/2) > 0$ as $\theta_i \in [-\pi, \pi]$). The algorithm consists of the following steps: first, map the input vector $\sum_i \alpha_i |v_i\rangle$ to $\sum_i \alpha_i |Qv_i\rangle$ by applying Q; then, use phase estimation as in theorem 3 with unitary W to compute an estimate of the eigenvalues θ_i and hence of the singular values $\sigma_i = \|A\|_F \cos(\theta_i/2)$; and finally undo Q to recover the state $\sum_i \alpha_i |v_i\rangle |\sigma_i\rangle$. This procedure is described in algorithm 1.

It remains to show how to construct the mappings P, Q and the unitary W that satisfy all the properties mentioned above that are required for the quantum singular value estimation algorithm.

▶ **Lemma 12.** *Let $A \in \mathbb{R}^{m\times n}$ be a matrix with singular value decomposition $A = \sum_i \sigma_i u_i v_i^t$ stored in the data structure in theorem 10. Then, there exist matrices $P \in \mathbb{R}^{mn\times m}, Q \in \mathbb{R}^{mn\times n}$ such that*

- **(i)** *The matrices P, Q are a factorization of A, i.e. $\frac{A}{\|A\|_F} = P^tQ$. Moreover, $P^tP = I_m$, $Q^tQ = I_n$, and multiplication by P, Q, i.e. the mappings $|y\rangle \to |Py\rangle$ and $|x\rangle \to |Qx\rangle$ can be performed in time $O(\text{polylog}(mn))$.*
- **(ii)** *The unitary $W = U \cdot V$, where U, V are the reflections $U = 2PP^t - I_{mn}$ and $V = 2QQ^t - I_{mn}$ can be implemented in time $O(\text{polylog}(mn))$.*
- **(iii)** *The isometry $Q : \mathbb{R}^n \to \mathbb{R}^{mn}$ maps a row singular vector v_i of A with singular value σ_i to an eigenvector Qv_i of W with eigenvalue $e^{\iota\theta_i}$ such that $\cos(\theta_i/2) = \sigma_i/\|A\|_F$.*

Proof. Let $P \in \mathbb{R}^{mn\times m}$ be a matrix with column vectors $e_i \otimes \frac{A_i}{\|A_i\|}$ for $i \in [m]$. In quantum notation multiplication by P can be expressed as

$$|Pe_i\rangle = |i, A_i\rangle = \frac{1}{\|A_i\|} \sum_{j\in[n]} A_{ij} |i, j\rangle , \quad \text{for } i \in [m].$$

Let $\widetilde{A} \in \mathbb{R}^m$ be the vector of Frobenius norms of the rows of the matrix A, that is $\widetilde{A}_i = \|A_i\|$ for $i \in [m]$. Let $Q \in \mathbb{R}^{mn \times n}$ be a matrix with column vectors $\frac{\widetilde{A}}{\|A\|_F} \otimes e_j$ for $j \in [n]$. In quantum notation multiplication by Q can be expressed as

$$|Qe_j\rangle = |\widetilde{A}, j\rangle = \frac{1}{\|A\|_F} \sum_{i \in [m]} \|A_i\| \, |i, j\rangle \,, \qquad \text{for } j \in [n].$$

The factorization $A = P^t Q$ follows easily by expressing the matrix product in quantum notation,

$$(P^t Q)_{ij} = \langle i, A_i | \widetilde{A}, j\rangle = \frac{\|A_i\|}{\|A\|_F} \frac{A_{ij}}{\|A_i\|} = \frac{A_{ij}}{\|A\|_F}.$$

The columns of P, Q are orthonormal by definition so $P^t P = I_m$ and $Q^t Q = I_n$. Multiplication by P and Q can be implemented in time polylog(mn) using quantum access to the data structure from theorem 10,

$$\begin{aligned} |y\rangle \to |y, 0^{\lceil \log n \rceil}\rangle = \sum_{i \in [m]} y_i \, |i, 0^{\lceil \log n \rceil}\rangle \xrightarrow{\widetilde{U}} \sum_{i \in [m]} y_i \, |i, A_i\rangle = |Py\rangle \\ |x\rangle \to |0^{\lceil \log m \rceil}, x\rangle = \sum_{j \in [n]} x_j \, |0^{\lceil \log m \rceil}, j\rangle \xrightarrow{\widetilde{V}} \sum_{j \in [n]} x_j \, |\widetilde{A}, j\rangle = |Qx\rangle \,. \end{aligned} \tag{12}$$

To show (ii), note that the unitary U is a reflection in $Col(P)$ and can be implemented as $U = \widetilde{U} R_1 \widetilde{U}^{-1}$ where $\widetilde{U}$ is the unitary in first line of equation (12) and R_1 is the reflection in the space $|y, 0^{\lceil \log n \rceil}\rangle$ for $y \in \mathbb{R}^m$. It can be implemented as a reflection conditioned on the second register being in state $|0^{\lceil \log n \rceil}\rangle$. The unitary V is a reflection in $Col(Q)$ and can be implemented analogously as $V = \widetilde{V} R_0 \widetilde{V}^{-1}$ where $\widetilde{V}$ is the unitary in the second line of equation (12) and R_0 is the reflection in the space $|0^{\lceil \log m \rceil}, x\rangle$ for $x \in \mathbb{R}^n$.

It remains to show that Qv_i is an eigenvector for W with eigenvalue $e^{\iota \theta_i}$ such that $\cos(\theta_i/2) = \sigma_i / \|A\|_F$. For every pair of singular vectors (u_i, v_i) of A, we define the two dimensional subspaces $\mathcal{W}_i = Span(Pu_i, Qv_i)$ and let $\theta_i/2 \in [-\pi/2, \pi/2]$ be the angle between Pu_i and $\pm Qv_i$. Note that $\mathcal{W}_i$ is an eigenspace for W which acts on it as a rotation by $\pm\theta_i$, since W is a reflection in the column space of Q followed by a reflection in the column space of P. Moreover, the relation $\cos(\theta_i/2) = \sigma_i / \|A\|_F$ is a consequence of the factorization in lemma 12, since we have

$$PP^t Qv_i = \frac{PAv_i}{\|A\|_F} = \frac{\sigma_i}{\|A\|_F} Pu_i \quad \text{and} \quad QQ^t Pu_i = \frac{QA^t u_i}{\|A\|_F} = \frac{\sigma_i}{\|A\|_F} Qv_i. \tag{13}$$

◀

Using the primitives from the preceding lemma, we next describe the singular value estimation algorithm and analyze it to prove theorem 11.

5.2.0.1 Analysis

The phase estimation procedure [12] with unitary W and precision parameter ϵ on input $|Qv_i\rangle$ produces an estimate such that $|\overline{\theta_i} - \theta_i| \leq 2\epsilon$. The estimate for the singular value is $\overline{\sigma_i} = \cos(\overline{\theta_i}/2)\|A\|_F$. The error in estimating $\sigma_i = \cos(\theta_i/2)\|A\|_F$ can be bounded as follows,

$$|\overline{\sigma_i} - \sigma_i| = |\cos(\theta_i/2) - \cos(\overline{\theta_i}/2)| \|A\|_F \leq \sin(\phi) \frac{|\overline{\theta_i} - \theta_i|}{2} \|A\|_F \leq \epsilon \|A\|_F \tag{14}$$

Algorithm 1 Quantum singular value estimation

Require: $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$ in the data structure from theorem 10, precision parameter $\epsilon > 0$.

1. Create $|x\rangle = \sum_i \alpha_i |v_i\rangle$.
2. Append a first register $|0^{\lceil \log m \rceil}\rangle$ and create the state $|Qx\rangle = \sum_i \alpha_i |Qv_i\rangle$ as in eq. (12).
3. Perform phase estimation with precision parameter $2\epsilon > 0$ on the input $|Qx\rangle$ for the unitary $W = U \cdot V$ where U, V are the unitaries in lemma 12 and obtain $\sum_i \alpha_i |Qv_i, \overline{\theta_i}\rangle$.
4. Compute $\overline{\sigma_i} = \cos(\overline{\theta_i}/2)\|A\|_F$ where $\overline{\theta_i}$ is the estimate from phase estimation, and uncompute the output of the phase estimation.
5. Apply the inverse of the transformation in step 2 to obtain $\sum_i \alpha_i |v_i\rangle |\overline{\sigma_i}\rangle$.

where $\phi \in [\theta_i/2 - \epsilon, \theta_i/2 + \epsilon]$. Algorithm 1 therefore produces an additive error $\epsilon\|A\|_F$ estimate of the singular values, the running time is $O(\text{polylog}(mn)/\epsilon)$ by theorem 3 as the unitary W is implemented in time $O(\text{polylog}(mn))$ by lemma 12. This concludes the proof of theorem 11.

One can define an algorithm for singular value estimation with input $|y\rangle = \sum_i \beta_i |u_i\rangle$. where u_i are the column singular vectors, by using the operator P from lemma 12 instead of Q in algorithm 1. The correctness follows from the same argument as above.

5.3 Quantum projection with threshold

Let $A = \sum_i \sigma_i u_i v_i^t$. We recall that $A_{\geq\sigma} = \sum_{\sigma_i \geq \sigma} \sigma_i u_i v_i^t$ is the projection of the matrix A onto the space spanned by the singular vectors whose singular values are bigger than σ. Also, $A_{\geq\sigma,\kappa}$ is the projection of the matrix A onto the space spanned by the union of the singular vectors whose corresponding singular values is bigger than σ and some subset of singular vectors whose corresponding singular values are in the interval $[(1-\kappa)\sigma, \sigma)$.

Algorithm 2 presents a quantum algorithm that given access to vector state x, a matrix A and parameters σ, κ, outputs the state $|A^+_{\geq\sigma,\kappa} A_{\geq\sigma,\kappa} x\rangle$, namely the projection of x onto the subspace spanned by the union of the row singular vectors whose corresponding singular values are bigger than σ and some subset of row singular vectors whose corresponding singular values are in the interval $[(1-\kappa)\sigma, \sigma)$.

For simplicity, we present the algorithm without a stopping condition and we will compute the expected running time. By stopping the algorithm after a number of iterations which is $\log(n)$ times more than the expected one, we can easily construct an algorithm with worst-case running time guarantees and whose correctness probability has only decreased by a factor of $(1 - 1/\text{poly}(n))$.

Let $\{v_i\}$ denote an orthonormal basis for $\mathbb{R}^n$ that includes all row singular vectors of the matrix A. We think of κ as a constant, for example $1/3$.

For the running time, note that the singular value estimation takes time $O(\text{polylog}(mn)/\epsilon)$, while the probability we obtain $|0\rangle$ in step 5 is $\frac{||A^+_{\geq\sigma,\kappa} A_{\geq\sigma,\kappa} x||^2}{\|x\|^2} \geq \frac{||A^+_{\geq\sigma} A_{\geq\sigma} x||^2}{\|x\|^2}$.

▶ **Theorem 13.** *Algorithm 2 outputs $|A^+_{\geq\sigma,\kappa} A_{\geq\sigma,\kappa} x\rangle$ with probability at least $1 - 1/\text{poly}(n)$ and in expected time $O(\frac{\text{polylog}(mn)\|A\|_F \|x\|^2}{\sigma ||A_{\geq\sigma} A^+_{\geq\sigma} x||^2})$.*

It is important to notice that the running time of the quantum projection algorithm depends only on the threshold σ (which we will take to be of the order $\frac{\|A\|_F}{\sqrt{k}}$) and not on

Algorithm 2 Quantum projection with threshold

Require: $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$ in the data structure from Theorem 10; parameters $\sigma, \kappa > 0$.

1. Create $|x\rangle = \sum_i \alpha_i |v_i\rangle$.
2. Apply the singular value estimation on $|x\rangle$ with precision $\epsilon \hat{=} \frac{\kappa}{2}\frac{\sigma}{\|A\|_F}$ to obtain the state
$$\sum_i \alpha_i |v_i\rangle |\overline{\sigma}_i\rangle$$
3. Apply on a second new register the unitary V that maps $|t\rangle|0\rangle \mapsto |t\rangle|1\rangle$ if $t < \sigma - \frac{\kappa}{2}\sigma$ and $|t\rangle|0\rangle \mapsto |t\rangle|0\rangle$ otherwise, to get the state
$$\sum_{i\in S} \alpha_i |v_i\rangle |\overline{\sigma}_i\rangle |0\rangle + \sum_{i\in\overline{S}} \alpha_i |v_i\rangle |\overline{\sigma}_i\rangle |1\rangle ,$$
where S is the union of all i's such that $\sigma_i \geq \sigma$ and some i's with $\sigma_i \in [(1-\kappa)\sigma, \sigma)$.
4. Apply the singular value estimation on the above state to erase the second register
$$\sum_{i\in S} \alpha_i |v_i\rangle |0\rangle + \sum_{i\in\overline{S}} \alpha_i |v_i\rangle |1\rangle = \beta |A^+_{\geq\sigma,\kappa} A_{\geq\sigma,\kappa} x\rangle |0\rangle + \sqrt{1-|\beta|^2}\, |A^+_{\geq\sigma,\kappa} A_{\geq\sigma,\kappa} x\rangle^{\perp} |1\rangle ,$$
with $\beta = \frac{||A^+_{\geq\sigma,\kappa} A_{\geq\sigma,\kappa} x||}{\|x\|}$.
5. Measure the second register in the standard basis. If the outcome is $|0\rangle$, output the first register and exit. Otherwise repeat step 1.

the condition number of A which may be very large. We will also show in the next section that in the recommendation systems, for most users the ratio $\frac{||A^+_{\geq\sigma} A_{\geq\sigma} x||^2}{\|x\|^2}$ is constant. This will conclude the analysis and show that the running time of the quantum recommendation system is polynomial in k and polylogarithmic in the matrix dimensions.

One could also use amplitude amplification to improve the running time of algorithm 2, once a careful error analysis is performed as the reflections are not exact. As we will see that the $\frac{||A^+_{\geq\sigma} A_{\geq\sigma} x||^2}{\|x\|^2}$ is constant for most users, this will not change asymptotically the running time of the algorithm and hence we omit the analysis.

6 Quantum recommendation systems

We have all the necessary ingredients to describe the quantum algorithm that provides good recommendations for a user i and that runs in time polylogarithmic in the dimensions of the preference matrix and polynomial in the rank k. As we said, in recommendation systems we assume that for the matrix T we have $\|T - T_k\|_F \leq \epsilon ||T||_F$ for some small approximation parameter ϵ and small rank k (no more than 100). In Algorithm 3, as in the classical recommendation systems, we assume we know k but in fact we just need to have a good estimate for it.

Note again that we do not put a stopping condition to the algorithm and we compute the expected running time. Again we can turn this into an algorithm with worst-case running time guarantees by stopping after running for $\log(n)$ times more than the expected running time, and the correctness probability has only decreased by a factor of $1 - 1/\text{poly}(n)$.

Algorithm 3 Quantum recommendation algorithm.

Require: A subsample matrix $\widehat{T} \in \mathbb{R}^{m\times n}$ (with sampling probability p) stored in the data structure from Theorem 10 and satisfying the conditions in Theorem 9; a user index i.

1: Apply the quantum projection procedure 2 with the matrix $\widehat{T}$, the vector corresponding to the i-th row $\widehat{T}_i$, with $\sigma = \sqrt{\frac{\epsilon^2 p}{2k}}\|\widehat{T}\|_F$ and $\kappa = 1/3$.
The algorithm runs in expected time $O(\text{polylog}(mn)\sqrt{k}\|\widehat{T}_i\|^2/\sqrt{p}\|\widehat{T}^+_{\geq\sigma}\widehat{T}_{\geq\sigma}\widehat{T}_i\|^2)$ and returns with probability at least $1 - 1/\text{poly}(n)$ the state $|\widehat{T}^+_{\geq\sigma,\kappa}\widehat{T}_{\geq\sigma,\kappa}\widehat{T}_i\rangle$.

2: Measure the above state in the computational basis to get a product j.

6.1 Analysis

Correctness

Let us check the correctness of the algorithm. Note that $\widehat{T}^+_{\geq\sigma,\kappa}\widehat{T}_{\geq\sigma,\kappa}\widehat{T}_i = (\widehat{T}_{\geq\sigma,\kappa})_i$, i.e. the i-th row of the matrix $\widehat{T}_{\geq\sigma,\kappa}$. Hence, the quantum projection procedure outputs with probability at least $1 - 1/\text{poly}(n)$ the state $|(\widehat{T}_{\geq\sigma,\kappa})_i\rangle$, meaning that our quantum recommendation algorithm with high probability outputs a product by sampling the i-th row of the matrix $\widehat{T}_{\geq\sigma,\kappa}$.

By Theorem 9, and by setting the parameters appropriately to get equation (11), we have that with probability at least $1 - exp(-19(\log n)^4)$,

$$\|T - \widehat{T}_{\geq\sigma,\kappa}\|_F \leq 9\epsilon\|T\|_F.$$

In this case, we can apply Theorem 6 with matrix $\widetilde{T} = \widehat{T}_{\geq\sigma,\kappa}$ to show that there exists a subset of users S' of size at least $(1-\delta-\zeta)m$ (for $\delta > 0$), such that on average over the users in S', the probability that our quantum algorithm provides a bad recommendation is

$$\Pr_{i\sim\mathcal{U}_{S'},j\sim(\widehat{T}_{\geq\sigma,\kappa})_i}[(i,j) \text{ bad}] \leq \frac{\left(\frac{9\epsilon(1+9\epsilon)}{1-9\epsilon}\right)^2}{\left(1/\sqrt{1+\gamma} - 9\epsilon/\sqrt{\delta}\right)^2 (1-\delta-\zeta)}.$$

Expected running time

We prove the following theorem

▶ **Theorem 14.** *For at least $(1-\xi)(1-\delta-\zeta)m$ users in the subset S', we have that the expected running time of Algorithm 3 is $O(\text{polylog}(mn)\text{poly}(k))$.*

Proof. First, by the conditions of Theorem 9, we must have $p \geq \frac{36\sqrt{2}(nk)^{1/2}}{\|A\|_F\epsilon^3}$. That is the theorem works even for a p which is sub-constant. However, in order to have the desired running time, we need to take p to be some constant, meaning that we need to subsample a constant fraction of the matrix elements. This is also the case for classical recommendation systems [5, 8].

Second, we need to show that for most users the term $W_i \equiv \frac{||\widehat{T}_i||^2}{||(\widehat{T}_{\geq\sigma,\kappa})_i||^2}$ that appears in the running time of the quantum projection algorithm is a constant. This is to be expected, since most typical rows of the matrix project very well onto the space spanned by the top singular vectors, since the spectrum of the matrix is well concentrated on the space of the top singular vectors.

As in Theorem 6, we focus on the users in the subset S', with $|S'| \geq (1-\delta-\zeta)m$, for which equations 4 and 6 hold. For these users we can use equation 6 with the matrix $\widetilde{T} = \widehat{T}_{\geq\sigma,\kappa}$ and error 9ϵ. We have

$$\begin{aligned} E_{i\in S'}[W_i] &= E_{i\in S'}[\frac{||\widehat{T}_i||^2}{||(\widehat{T}_{\geq\sigma,\kappa})_i||^2}] \leq \frac{E_{i\in S'}[||\widehat{T}_i||^2]}{\frac{||\widehat{T}||_F^2}{(1+\epsilon)^2 m}\left(\frac{1}{\sqrt{1+\gamma}}-\frac{9\epsilon}{\sqrt{\delta}}\right)^2} \leq \frac{\frac{||\widehat{T}||_F^2}{(1-\delta-\zeta)m}}{\frac{||\widehat{T}||_F^2}{(1+\epsilon)^2 m}\left(\frac{1}{\sqrt{1+\gamma}}-\frac{9\epsilon}{\sqrt{\delta}}\right)^2} \\ &\leq \frac{(1+\epsilon)^2}{(1-\delta-\zeta)\left(\frac{1}{\sqrt{1+\gamma}}-\frac{9\epsilon}{\sqrt{\delta}}\right)^2}. \end{aligned}$$

By Markov's inequality, for at least $(1-\xi)|S'|$ users in S' we have $W_i \leq \frac{(1+\epsilon)^2}{\xi(1-\delta-\zeta)\left(\frac{1}{\sqrt{1+\gamma}}-\frac{9\epsilon}{\sqrt{\delta}}\right)^2}$, which for appropriate parameters is a constant. Hence, for at least $(1-\xi)(1-\delta-\zeta)m$ users, the quantum recommendation algorithm has an expected running time of $O(\text{poly}(k)\text{polylog}(mn))$ and produces good recommendations with high probability. As we said we can easily turn this into a worst-case running time, by stopping after running $\log(n)$ times more than the expected running time and hence decreasing the correctness only by a factor of $1 - 1/\text{poly}(n)$. ◀

References

1. Scott Aaronson. Read the fine print. *Nature Physics*, 11(4):291–293, 2015.
2. Dimitris Achlioptas and Frank McSherry. Fast computation of low rank matrix approximations. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 611–618. ACM, 2001.
3. Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.
4. Baruch Awerbuch, Boaz Patt-Shamir, David Peleg, and Mark Tuttle. Improved recommendation systems. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1174–1183. Society for Industrial and Applied Mathematics, 2005.
5. Yossi Azar, Amos Fiat, Anna Karlin, Frank McSherry, and Jared Saia. Spectral analysis of data. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 619–626. ACM, 2001.
6. Robert M Bell and Yehuda Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.
7. Andrew M Childs. On the relationship between continuous-and discrete-time quantum walk. *Communications in Mathematical Physics*, 294(2):581–603, 2010.
8. Petros Drineas, Iordanis Kerenidis, and Prabhakar Raghavan. Competitive recommendation systems. In *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, pages 82–90. ACM, 2002.
9. Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv preprint quant-ph/0208112*, 2002.
10. Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.
11. Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM, 2013.
12. A Yu Kitaev. Quantum measurements and the abelian stabilizer problem. *arXiv preprint quant-ph/9511026*, 1995.

13 Yehuda Koren and Robert Bell. Advances in collaborative filtering. In *Recommender systems handbook*, pages 145–186. Springer, 2011.

14 Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

15 Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning. *Arxiv preprint:1307.0411*, 2013.

16 Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum self analysis. *Arxiv preprint:1307.1401*, 2013.

17 Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum support vector machine for big feature and big data classification. *Arxiv preprint:1307.0471*, 2013.

18 Christos H Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 159–168. ACM, 1998.

19 Anupam Prakash. Quantum algorithms for linear algebra and machine learning. *Ph.D Thesis, University of California, Berkeley.*, 2014.

20 Anand Rajaraman and Jeffrey D Ullman. *Mining of massive datasets*, volume 77. Cambridge University Press Cambridge, 2012.

21 Elaine Rich. User modeling via stereotypes*. *Cognitive science*, 3(4):329–354, 1979.

22 Cyril Stark. Recommender systems inspired by the structure of quantum theory. *arXiv preprint arXiv:1601.06035*, 2016.

A The data structure

We prove the following theorem.

▶ **Theorem 15.** *(Theorem 10 restated) Let $A \in \mathbb{R}^{m\times n}$ be a matrix. Entries (i, j, A_{ij}) arrive in the system in some arbitrary order, and w denotes the number of entries that have already arrived in the system. There exists a data structure to store the matrix A with the following properties:*

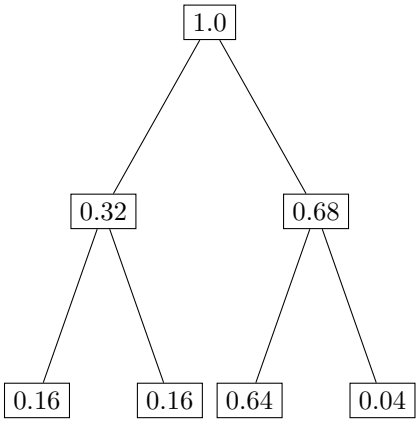
(i) *The size of the data structure is $O(w \log^2(mn))$.*

(ii) *The time to store a new entry (i, j, A_{ij}) is $O(\log^2(mn))$.*

(iii) *A quantum algorithm that has quantum access to the data structure can perform the mapping $\widetilde{U} : |i\rangle|0\rangle \to |i\rangle|A_i\rangle$, for $i \in [m]$, corresponding to the rows of the matrix currently stored in memory and the mapping $\widetilde{V} : |0\rangle|j\rangle \to |\widetilde{A}\rangle|j\rangle$, for $j \in [n]$, where $\widetilde{A} \in \mathbb{R}^m$ has entries $\widetilde{A}_i = \|A_i\|$ in time* $\mathrm{polylog}(mn)$.

Proof. The data structure consists of an array of m binary trees $B_i, i \in [m]$. The trees B_i are initially empty. When a new entry (i, j, A_{ij}) arrives the leaf node j in tree B_i is created if not present and updated otherwise. The leaf stores the value A_{ij}^2 as well as the sign of A_{ij}. The depth of each tree B_i is at most $\lceil \log n \rceil$ as there can be at most n leaves. An internal node v of B_i stores the sum of the values of all leaves in the subtree rooted at v, i.e. the sum of the square amplitudes of the entries of A_i in the subtree. Hence, the value stored at the root is $\|A_i\|^2$. When a new entry arrives, all the nodes on the path from that leaf to the tree root are also updated. The different levels of the tree B_i are stored as ordered lists so that the address of the nodes being updated can be retrieved in time $O(\log mn)$. The binary tree for a 4-dimensional unit vector for which all entries have arrived is illustrated in figure 1.

The time required to store entry (i, j, A_{ij}) is $O(\log^2 mn)$ as the insertion algorithm makes at most $\lceil \log n \rceil$ updates to the data structure and each update requires time $O(\log mn)$ to retrieve the address of the updated node.



Let $|\phi\rangle = 0.4\,|00\rangle + 0.4\,|01\rangle + 0.8\,|10\rangle + 0.2|11\rangle$.

- Rotation on qubit 1:
  $|0\rangle\,|0\rangle \to (\sqrt{0.32}\,|0\rangle + \sqrt{0.68}\,|1\rangle)\,|0\rangle$
- Rotation on qubit 2 conditioned on qubit 1:

$$(\sqrt{0.32}\,|0\rangle + \sqrt{0.68}\,|1\rangle)\,|0\rangle \to$$
$$\sqrt{0.32}\,|0\rangle\,\frac{1}{\sqrt{0.32}}(0.4\,|0\rangle + 0.4\,|1\rangle) +$$
$$\sqrt{0.68}\,|1\rangle\,\frac{1}{\sqrt{0.68}}(0.8\,|0\rangle + 0.2\,|1\rangle)$$

Figure 1 Vector state preparation illustrated for 4-dimensional state $|\phi\rangle$.

The memory requirement for the data structure is $O(w \log^2 mn)$ as for each entry (i, j, A_{ij}) at most $\lceil \log n \rceil$ new nodes are added, each node requiring $O(\log mn)$ bits.

We now show how to perform $\widetilde{U}$ in time polylog(mn) if an algorithm has quantum access to this classical data structure. The state preparation procedure using pre-computed amplitudes is well known in the literature, for instance see [9]. The method is illustrated for a 4-dimensional state $|\phi\rangle$ corresponding to a unit vector in figure 1. The amplitudes stored in the internal nodes of B_i are used to apply a sequence of conditional rotations to the initial state $|0\rangle^{\lceil \log n \rceil}$ to obtain $|A_i\rangle$. Overall, there are $\lceil \log n \rceil$ rotations applied and for each one of them we need two quantum queries to the data structure (from each node in the superposition we query its two children).

The amplitude stored at an internal node of B_i at depth t corresponding to $k \in \{0,1\}^t$ is,

$$B_{i,k} := \sum_{j \in [n], j_{1:t} = k} A_{ij}^2$$

where $j_{1:t}$ denotes the first t bits in the binary representation for j. Note that $B_{i,k}$ is the probability of observing outcome k if the first t bits of $|A_i\rangle$ are measured in the standard basis. Conditioned on the first register being $|i\rangle$ and the first t qubits being in state $|k\rangle$ the rotation is applied to the $(t+1)$ qubit as follows

$$|i\rangle\,|k\rangle\,|0\rangle \to |i\rangle\,|k\rangle\,\frac{1}{\sqrt{B_{i,k}}}\left(\sqrt{B_{i,k0}}\,|0\rangle + \sqrt{B_{i,k1}}\,|1\rangle\right).$$

The sign is included for rotations applied to the $\lceil \log n \rceil$-th qubit

$$|i\rangle\,|k\rangle\,|0\rangle \to |i\rangle\,|k\rangle\,\frac{1}{\sqrt{B_{i,k}}}\left(sgn(A_{k0})\sqrt{B_{i,k0}}\,|0\rangle + sgn(A_{k1})\sqrt{B_{i,k1}}\,|1\rangle\right).$$

Last, we show how to perform $\widetilde{V}$ in time polylog(mn). Note that the amplitudes of the vector $\widetilde{A}$ are equal to $\|A_i\|$, and the values stored on the roots of the trees B_i are equal to $\|A_i\|^2$. Hence, by a similar construction (another binary tree) for the m roots, we can perform the unitary $\widetilde{V}$ efficiently.

◀

Random Walks in Polytopes and Negative Dependence

Yuval Peres[1], Mohit Singh[2], and Nisheeth K. Vishnoi[3]

1 **Microsoft Research, Redmond, USA**
peres@microsoft.com
2 **Georgia Institute of Technology, Atlanta, USA**
mohitsinghr@gmail.com
3 **Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland**
nisheeth.vishnoi@gmail.com

Abstract

We present a Gaussian random walk in a polytope that starts at a point inside and continues until it gets absorbed at a vertex. Our main result is that the probability distribution induced on the vertices by this random walk has strong negative dependence properties for matroid polytopes. Such distributions are highly sought after in randomized algorithms as they imply concentration properties. Our random walk is simple to implement, computationally efficient and can be viewed as an algorithm to round the starting point in an unbiased manner. The proof relies on a simple inductive argument that synthesizes the combinatorial structure of matroid polytopes with the geometric structure of multivariate Gaussian distributions. Our result not only implies a long line of past results in a unified and transparent manner, but also implies new results about constructing negatively associated distributions for *all* matroids.

1998 ACM Subject Classification F.2.1 Numerical Algorithms and Problems

Keywords and phrases Random walks, Matroid, Polytope, Brownian motion, Negative dependence

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.50

1 Introduction

A basic problem that underlies a large number of approximation algorithms for combinatorial problems is: given a fractional point, how should it be rounded to an integral point? Typically, the fractional point is obtained by solving a linear program and has nice properties which one would like the integeral point to inherit. For instance, the fractional point might belong to the spanning tree polytope of a graph and the goal might be to round the point to a spanning tree whose cost is not much more and, in addition, satisfies some constraints satisfied by the fractional point; e.g., number of edges in the tree across each cut is small. A common approach towards this is to construct randomized rounding algorithms for such a problem which output an unbiased distribution over the underlying set of integral objects (spanning trees for the above example). Since in most interesting cases the set of integral objects is not a box, the distributions have correlations. These correlations end up being quite problematic when it comes to ensure that with high probability the integral point also satisfies the *additional* constraints the fractional point satisfies. The ultimate hope here is that the distribution essentially behaves like a product distribution so that one can apply concentration results such as Chernoff bounds. On the one hand, this has recently led TCS researchers to come up with a wide variety of ingenious rounding algorithms which have resulted in significant


© Yuval Peres, Mohit Singh and Nisheeth K. Vishnoi;
licensed under Creative Commons License CC-BY

progress on fundamental algorithmic problems [11, 2, 6, 7, 12, 19, 1] (see the book [8] for more examples). On the other hand, mathematicians have been investigating what kind of negative dependence properties result in the phenomena of measure concentration [17, 3]. Typically, it is non-trivial to both compute these distributions and prove their negative dependence properties.

In this paper we take a geometric approach and propose a very simple rounding algorithm and prove that it gives an unbiased distribution with *negative association* property for *all* matroids. Negative association is a significant strengthening of the negative correlation property. Roughly a set of random variables $(X_1, \ldots, X_n)$ is said to be negatively associated if for any two monotone functions f and g which act on distinct sets of coordinates S and T respectively, the corresponding random variables $f(X_i)_{i\in S}$ and $g(X_j)_{j\in T}$ are negatively correlated; see Definition 1 and Theorem 4.

A bit more formally, given a matroid over a universe of size n, consider the polytope in $\mathbb{R}^n$ that is the convex hull of all the bases of the matroid. Let θ be a given fractional point that lies inside this polytope and is to be rounded to one of the bases. Our algorithm starts at θ and keeps taking small Gaussian steps centered at the current point – thus maintaining the unbiasedness. If at some point the trajectory hits a constraint on the boundary of the polytope it never leaves it – meaning that the Gaussian in the next step is chosen so as to have no mass outside of this subspace. Thus, eventually, the trajectory gets absorbed at a vertex of the polytope. This rounding algorithm is inspired by the work of [15] on discrepancy and its simplicity is self-evident. At any given time, the algorithm has to keep track of the tight constraints and compute a Gaussian in the space corresponding to the intersection of these constraints.

Our proof synthesizes polyhedral properties about matroids with well-known properties of Gaussian distribution. Our key structural observation is that if F is a d-dimensional face of a matroid polytope and Σ_F is the covariance matrix of the d-dimensional Gaussian obtained by orthogonally projecting an n-dimensional Gaussian onto F, then all the off-diagonal entries of Σ are non-positive; see Theorem 5. The proof of this relies on an elementary *uncrossing* argument from matroid theory. *Thus, Gaussian distributions on faces of matroid polytopes have the negative correlation property.* This, in turn, immediately leads to an inductive argument that shows that from the beginning to the end of this random walk, the distribution has the negative correlation property. Finally, to go from negative correlation of Gaussian distributions to negative association, we employ a result that implies that for Gaussian distribution negative correlations implies negative association [13]. Roughly, this is a manifestation of the fact that all moments of Gaussian distributions are completely determined by their covariance matrix.

Negative dependence properties for distributions on bases of matroid have been extensively studied. Prior to our work, negatively associated distributions were known only for *balanced* matroids. In particular, [10] shows that the uniform distribution is negatively associated for balanced matroids and gives a Markov chain Monte Carlo method to sample from such a distribution. Unfortunately, the uniform, or more generally, an entropy maximizing distribution on general matroids does not give negative association property. For general matroids, the weaker negative cylindrical property (see Definition 3) was shown for the *pipage rounding* and *randomized swap rounding* [6]. The other two matroids that have been extensively studied are the uniform matroid, where bases are all sets of size k for integer k [20, 5] and the partition matroid [9, 8].

Lovett and Meka [15], who studied this random walk on the polytope obtained by intersecting the hypercube with few *discrepancy* constraints, were interested in the number

of integral coordinates of the output vertex. In our setting, we study the walk on integral polytopes, convex hull of bases of matroids, which are typically defined by exponentially many constraints and are interested in negative dependence properties of the output distribution.

While both the algorithm and the proof are simple and the reason for negative dependence is clear, several problems about the distribution remain open. First and foremost, what can we say about non-matroid polytopes? Do non-Gaussian distributions help? While we can understand the random walk locally, a global perspective seems hard. Concretely, can we prove that the distribution obtained by our random walk is the solution to some optimization problem? Finally, do much stronger forms of negative dependence, for example, Strongly Rayleigh property [3] holds for these distributions in specific matroids (see also [4])?

1.1 Preliminaries

We first give the following definition.

▶ **Definition 1** (Negative Association)**.** Let $X_1, \ldots, X_n$ be boolean random variables. Then X_i's are *negatively associated* if for every non-decreasing functions $f : \{0,1\}^n \to \mathbb{R}$ and $g : \{0,1\}^n \to \mathbb{R}$ we have

$$E[f(X_1, \ldots, X_n)g(X_1, \ldots, X_n)] \leq E[f(X_1, \ldots, X_n)]E[g(X_1, \ldots, X_n)] \tag{1}$$

if f and g depend on disjoint sets of coordinates.

While negative association is hard to verify, under very mild assumptions, it implies versions of the central limit theorems (see Yuan et al [21] and Pattersen et al [16]).

A much weaker notion of negative dependence is the negative correlation defined below.

▶ **Definition 2.** Let $X_1, \ldots, X_n$ be real valued random variables. Then X_i's are *negatively correlated* if for each $i \neq j \in [n]$, we have

$$E[X_i X_j] \leq E[X_i]E[X_j]. \tag{2}$$

Another notion of negative dependence is the negative cylindrical property that is stronger than negative correlation but weaker than negative association. This property is also well studied since it is enough to imply tail bounds ala Chernoff bounds.

▶ **Definition 3.** Let $X_1, \ldots, X_n$ be real valued random variables. Then X_i's have the *negatively cylindrical property* if for each $S \subseteq [n]$, we have

$$E[\prod_{i \in S} X_i] \leq \prod_{i \in S} E[X_i]. \tag{3}$$

A set system $\mathcal{M} = (U, \mathcal{I})$ is called a matroid if $\mathcal{I} \subseteq 2^U$ satisfies two axioms (i) $A \in \mathcal{I}$ and $B \subseteq A$ implies that $B \in \mathcal{I}$, (ii) $A, B \in \mathcal{I}$ such that $|A| > |B|$ implies that there exists $a \in A \backslash B$ such that $B \cup \{a\} \in \mathcal{I}$. Sets of $\mathcal{I}$ are called *independent* sets and the maximal sets in $\mathcal{I}$ are called *bases* of matroid $\mathcal{M}$. For a matroid $\mathcal{M}$, the corresponding *matroid polytope* is the convex hull of the indicator vectors of all the bases of $\mathcal{M}$.

2 Our Algorithm and Result

2.1 Algorithm

We present a discrete implementation of the random walk algorithm. We first introduce some notation. Let

$$P = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq b\}$$

be the inequality description of P. Often we refer to individual inequalities or equalities as $\mathbf{a}_j^\top \mathbf{x} \leq b_j$ or $\mathbf{a}_j^\top \mathbf{x} = b_j$. For any face F of P defined by

$$F = \{\mathbf{x} \in P : A^= \mathbf{x} = b^=\},$$

let $d(F)$ denote the dimension of F. Let $C(F)$ denote the matrix whose columns form an orthonormal basis of the subspace

$$\{\mathbf{x} \in \mathbb{R}^n : A^= \mathbf{x} = 0\}.$$

Observe that the dimension of $C(F)$ is $n \times d(F)$.

Algorithm 1: Algorithm Random Walk

Input: θ, error parameter ε. Let $T = \frac{n^2}{\varepsilon^2}$.
Initialization $\mathbf{x}_0 \leftarrow \theta$. Let $t \leftarrow 0$, $F_0 = P$.
while dimension of $F_t > 0$ or $t \leq T$ **do**
  while there exists j such that $0 < b_j - \mathbf{a}_j^\top \mathbf{x}_t < n\varepsilon$, **do**
    Let $\mathbf{y}$ denote the point in $F_t \cap \{\mathbf{x}: \mathbf{a}_j^\top \mathbf{x} = b_j\}$ closest to $\mathbf{x}_t$.
    Let $\mathbf{x}_t \leftarrow \mathbf{y}$, $F_t \leftarrow F_t \cap \{\mathbf{x}: \mathbf{a}_j^\top \mathbf{x} = b_j\}$.
  end while
  Let d denote the dimension of F_t and let $\mathbf{g}$ be a normal Gaussian in d dimensions.
  Let $\mathbf{x}_{t+1} = \mathbf{x}_t + \varepsilon \cdot C(F_t)\mathbf{g}$ and let $F_{t+1} \leftarrow F_t$. If $\mathbf{x}_{t+1} \notin P$, then abort.
  Let $t \leftarrow t + 1$.
end while
Return $\mathbf{x}_t$.

If the algorithm ends at time $t^* \leq T$ with the vertex $\mathbf{x}_{t^*}$, we let $\mathbf{x}_t = \mathbf{x}_{t^*}$ for each $t^* \leq t \leq T$.

2.2 Main Result

Our main result is to show that the random walk algorithm described above gives a distribution over vertices of the matroid polytope that is negatively associated.

▶ **Theorem 4.** *Given any matroid $\mathcal{M} = (U, \mathcal{I})$, let P denote the convex hull of bases of $\mathcal{M}$ and let $n = |U|$. Given any $\theta \in P$ and error parameter $\varepsilon > 0$, the random walk algorithm returns a vertex of P before time $T = \frac{n^2}{\varepsilon^2}$ with probability at least $1 - e^{-n}$. Moreover, conditioned on algorithm returning a vertex of P, the output random vertex $\mathbf{x}_T$ satisfies the following properties.*

1. *For each $i \in U$,*

$$\theta(i) - n^2\varepsilon \leq E[\mathbf{x}_T(i)] \leq \theta(i) + n^2\varepsilon.$$

2. *For every 1-Lipschitz non-decreasing functions $f_1 : \mathbb{R}^U \to [0,1]$ and $f_2 : \mathbb{R}^U \to [0,1]$ depending on disjoint set of coordinates, we have*

$$E[f_1(\mathbf{x}_T) f_2(\mathbf{x}_T)] \leq E[f_1(\mathbf{x}_T)] E[f_2(\mathbf{x}_T)] + n\sqrt{n}\varepsilon \tag{4}$$

Moreover, the expected running time of the algorithm is polynomial in n and $\frac{1}{\varepsilon}$.

To prove the theorem, we rely on two crucial observations which also illustrate the crucial role of matroid polytopes. We show that for *every* face F of P, a standard Gaussian projected on F is negatively correlated; see Theorem 5. This result relies crucially on the facial structure of the matroid polytopes and a characterization of every face in terms of tight constraints defining the matroid polytope. Secondly, we use the classical result that for Gaussian random variables, negative correlation implies negative association (see Theorem 12). The above results use an inductive argument to show that the random walk algorithm leads to a distribution that is negatively associated.

▶ Remark. We mention that we describe the discrete version due to algorithmic implications but it naturally leads to error terms in Theorem 4. From a structural point of view, one can construct a Brownian motion that sticks to the face of the polytope as does our random walk and ends at a vertex of the polytope. The resulting distribution over bases of the polytope will satisfy the marginals and the inequality for negative association exactly.

3 Projected Gaussian in Matroids

In this section, we prove that standard Gaussian projected on any face of the matroid polytope is negatively correlated.

▶ **Theorem 5.** *Let $\mathcal{M} = (U, \mathcal{I})$ denote a matroid and P denote the convex hull of all bases of $\mathcal{M}$. Let F be the face of P and $g \in R^U$ be a standard Gaussian random variable. Then for any distinct $i, j \in U$ we have $E[(Cg)_i(Cg)_j] \leq 0$ where C is the projection matrix projecting onto F.*

Proof. Before we prove the general case, we prove the case when $\mathcal{M}$ is a uniform matroid. The convex hull of the bases of this matroid is well understood and has well characterized faces. The general case, whose facial structure is more complicated, will use this case as a building block.

Let k be an integer and $\mathcal{I} = \{S \subseteq U : |S| \leq k\}$. In this case, we have

$$P = \{x \in \mathbb{R}^U : \sum_{i \in U} x_i = k, \;\; 0 \leq x_i \leq 1 \;\; \forall i \in U\}.$$

First consider the face which is P itself. We let n denote the size of $|U|$.

▶ **Lemma 6.** *Let C denote the $n \times (n-1)$ matrix whose columns form the orthonormal basis of subspace $\{x \in \mathbb{R}^U : \sum_{i \in U} x_i = 0\}$. We have $CC^\top = I_n - \frac{1}{n} J_n$ where I_n denotes the identity matrix in n dimensions and J_n denotes the n dimensional matrix with all ones.*

Proof. Let $\hat{C}$ be $n \times n$ formed by adding the column $\frac{1}{\sqrt{n}}(1, \ldots, 1)$ to C. Then all columns of $\hat{C}$ form an orthonormal basis of $\mathbb{R}^n$. Then $\hat{C}\hat{C}^\top = I_n$. But $\hat{C}\hat{C}^\top = CC^\top + \frac{1}{n} J_n$ giving us that $CC^\top = I_n - \frac{1}{n} J_n$. ◀

Since all off-diagonal entries of $I_n - \frac{1}{n} J_n$ are negative, this implies that the standard Gaussian projected on P is negatively correlated. Any other face F is obtained by setting some of the variables to 0 or 1 in P. Thus the covariance matrix of the face will be a block matrix of the form $I_d - \frac{1}{n} J_d$ with the zero matrix. This completes the proof for the case when $\mathcal{M}$ is a uniform matroid.

We now consider the general case. Let $\mathcal{M} = (U, \mathcal{I})$ denote a matroid and P denote the convex hull of all independent sets of $\mathcal{M}$. In the next lemma, we characterize the possible covariance matrices for projections of normal Gaussians on any of the faces of the matroid

polytope. It shows that the covariance matrix is of block diagonal form where each block has the same structure as the covariance matrix for the uniform matroid as seen earlier.

▶ **Lemma 7.** *Let F be a face of P and X be a projection of a normal Gaussian on F. The covariance matrix of X (up to permuting indices of columns and rows) is $\Sigma = CC^\top$ where Σ is a block diagonal matrix with blocks of size $d_1, \ldots, d_r$ such that $\sum_i d_i = n$ and each block of size d_i is exactly $I_{d_i} - \frac{1}{d_i} J_{d_i}$ where I_{d_i} is the identity matrix and J_{d_i} is the all-ones matrix with sizes $d_i \times d_i$.*

Proof. The proof uses the characterization of any face of P by a maximal set of tight constraints which satisfy the chain structure. A collection of sets $A_1, A_2, \ldots, A_k \subseteq U$ form a *chain* if $A_1 \subseteq A_2 \subseteq \cdots \subseteq A_k$. The lemma is standard using the *uncrossing* technique (see Chapter 40 [18] or Lemma 5.2.4 [14]).

▶ **Claim 8.** *Given any face F of P, there exists a chain $S'_1 \subseteq S'_2 \subseteq \cdots \subseteq S'_r = U$ where $r = n - d$ and integers $k_j \geq 0$ for each $1 \leq j \leq r$ such that*

$$F = \{x \in P : \sum_{i \in S'_j} x_j = k_j \ \ \forall\, 1 \leq j \leq r\}.$$

Let $S_j = S'_j \setminus S'_{j-1}$ for each $1 \leq j \leq r$ where $S_0 = \emptyset$. Then the following two subspaces are equal

$$\{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}(S'_j) = 0 \ \ \forall\, 1 \leq j \leq r\} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}(S_j) = 0 \ \ \forall\, 1 \leq j \leq r\}.$$

Thus columns of C can be chosen to form an orthonormal basis of the subspace

$$\{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}(S_j) = 0 \ \ \forall\, 1 \leq j \leq r\}.$$

Since $\{S_j : 1 \leq j \leq r\}$ form a partition of U, we can choose columns of C to be divided into groups of size $|S_j| - 1$ for each $1 \leq j \leq r$ where the j^{th} group is supported on the elements of S_j. Let C_j denote the submatrix with these $|S_j| - 1$ columns. Thus, we can apply Lemma 6 and $C_j^\top C_j$ has the property that the non-zero entries form a block matrix of $I_{d_i} - \frac{1}{d_i} J_{d_i}$ on the elements corresponding to S_j. Since $C'_j s$ have disjoint support, we have $CC^\top = \sum_{i=1}^r C_j C_j^\top$ as required. ◀

This completes the proof of Theorem 5. ◀

4 Negative Dependence Properties

In this section, we show the negative dependence properties of the random walk algorithm in matroid polytopes.

First, we have the following simple claim.

▶ **Lemma 9.** *The algorithm ends in $T = \frac{n^2}{\varepsilon^2}$ iterations with probability at least $1 - e^{-n}$. Conditioned on terminating, if the algorithm terminates with solution x_T, then $x_T = \theta + \sum_{t=1}^T \varepsilon C(F_t) g_t + \sum_{i=1}^n \zeta_i$ where g_t are independent Gaussians and $|\zeta_i| \leq n\varepsilon$ for each i. Here $C(F_t)$ is the projection matrix at time t and is also a random variable.*

Proof. The probability that the algorithm aborts in a single iteration is bounded by the probability of the event that a Gaussian with covariance at most εI is outside the ball of radius εn. Standard concentration result that this probability is bounded by e^{-n^2}. Thus the probability of aborting in the first T iterations is at most $\frac{n^2}{\varepsilon^2} e^{-n^2} \leq \frac{1}{2} e^{-n}$ for large enough n

and $\varepsilon < \frac{1}{n^2}$. Let us condition on the event that the algorithm does not abort. Now we bound the number of iterations. In each iteration, the expected distance squared from θ increases by at least ε^2. This follows since

$$E[\|\mathbf{x}_{t+1} - \theta\|_2^2 | x_t] = \varepsilon^2 E[gC(F_t)C(F_t)^\top g] \geq \varepsilon^2$$

where we use the fact that $C(F_t)^\top g$ is projection of a standard Gaussian in at least dimension one. Taking expectation over x_t, we get the squared distance increases by at least ε. Since the distance is bounded by diameter of P which is $\sqrt{n}$. Thus, with probability at least $1 - \frac{1}{2}e^{-n}$, the algorithm terminates in $n^2\varepsilon^2$ iterations with a vertex of P conditioned on the event that it doesn't terminate. Thus, we get the first claim.

Now consider the case that the algorithm succeeds and returns x_T. At each iteration t, we add $\varepsilon C(F_t)g_t$ to the current vector except when we project on a face. Since the dimension of the face reduces by one, there are at most n projection steps. The error term introduced by the projection is denoted by ζ_i where $|\zeta_i| \leq n\varepsilon$ for each $1 \leq i \leq n$. thus we obtain that

$$\mathbf{x}_T = \theta + \sum_{t=1}^{T} \varepsilon C(F_t)g_t + \sum_{i=1}^{n} \zeta_i. \qquad \blacktriangleleft$$

From now, we condition on the event that the algorithm does not abort and terminates with a vertex at iteration T. Thus all expectations stated from now on are conditioned on this event.

4.1 Negative Cylinder Property

We first show the negative cylinder property as defined in Definition 3. Recall that negative cylinder property is enough to obtain concentration results as given by Chernoff bounds.

▶ **Lemma 10.** *For any subset $R \subseteq U$,*

$$E[\prod_{i\in R} \mathbf{x}_T(i)] \leq \prod_{i\in R} E[\mathbf{x}_T(i)] + 2n^2\varepsilon = \prod_{i\in R} \theta(i) + 2n^2\varepsilon$$

Proof. We will show that $\prod_{i\in R} \mathbf{x}_t(i)$ forms a supermartingale (modulo the error terms). Since $\prod_{i\in R} \mathbf{x}_0(i) = \prod_{i\in R} \theta(i)$, we will have the claim. Let us verify the supermartingale property, by first ignoring the error term introduced due to fixing a constraint. Recall that

$$\mathbf{x}_{t+1} = \mathbf{x}_t(i) + \varepsilon C(F_t)g_t.$$

Let $h_t = C(F_t)g_t$. Consider any time $t+1$.

$$\begin{aligned}
E[\prod_{i\in R} \mathbf{x}_{t+1}(i) | \mathbf{x}_t] &= E[\prod_{i\in R} (\mathbf{x}_t(i) + \varepsilon h_t) \,|\mathbf{x}_t] \\
&= \prod_{i\in R} \mathbf{x}_t(i) + \varepsilon \sum_{j\in R} E[h_t(j)|\mathbf{x}_t] \cdot \left(\prod_{i\in R\setminus\{j\}} \mathbf{x}_t(i) \right) \\
&\quad + \varepsilon^2 \sum_{j,k\in R, j\neq k} E[h_t(j)h_t(k)|\mathbf{x}_t] \cdot \left(\prod_{i\in R\setminus\{j,k\}} \mathbf{x}_t(i) \right) \\
&\quad + \varepsilon^3 \sum_{j,k,l\in R, j\neq k\neq l} E[h_t(j)h_t(k)h_t(l)|\mathbf{x}_t] \cdot \left(\prod_{i\in R\setminus\{j,k,l\}} \mathbf{x}_t(i) \right) + \dots \\
&\leq \prod_{i\in R} \mathbf{x}_t(i) + 2\varepsilon^3 n^3
\end{aligned}$$

where we use the fact that $E[h_t(j)|\mathbf{x}_t] = 0$ for each $j \in R$ and $\varepsilon^2 \sum_{j,k \in R, j \neq k} E[h_t(j)h_t(k)|\mathbf{x}_t] \leq 0$ for each $j \neq k \in R$ from Theorem 5. The later terms have increasing powers of ε and thus we can bound the error terms using the fact that $\varepsilon \leq \frac{1}{n^2}$.

Applying an inductive argument, and also incorporating the error term introduced due to fixing constraints, we obtain the lemma.

◀

4.2 Negative Association

We now prove the stronger negative dependence property of negative association. This relies on the following classical lemma about Gaussian distribution that says that pairwise negative correlation is enough to obtain negative association. The definition of negative association is extended to real random variables as well.

▶ **Definition 11** (Negative Association)**.** Let $X_1, \ldots, X_n$ be real valued random variables. Then X_i's are *negatively associated* if for every non-decreasing functions $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}$ we have

$$E[f(X_1, \ldots, X_n)g(X_1, \ldots, X_n)] \leq E[f(X_1, \ldots, X_n)]E[g(X_1, \ldots, X_n)] \tag{5}$$

if f and g depend on disjoint set of coordinates.

Observe that the definitions are consistent and if a set of boolean random variables are negatively associated as per Definition 1 then they are also negatively associated as per Definition 11. This follows since every non-decreasing $f : \{0,1\}^n \to \mathbb{R}$ which depends on $S \subseteq [n]$ can be extended to a non-decreasing function $\hat{f} : \mathbb{R}^n \to \mathbb{R}$ which depends on S. Indeed, for any $x \in [0,1]^n$, let

$$\hat{f}(x) = \sum_{T \subseteq [n]} f(T) \prod_{i \in T} x_i \prod_{i \notin T} (1 - x_i)$$

and for any other x, let

$$\hat{f}(x) = f(x \wedge 1)$$

where $(x \wedge 1)_i = \min\{x_i, 1\}$. A straightforward check shows that $\hat{f}$ and f agree on the boolean hypercube. Moreover, $\hat{f}$ is non-decreasing and only depends on S.

▶ **Theorem 12.** *[13] Let $f_1 : \mathbb{R}^n \to \mathbb{R}$ and $f_2 : \mathbb{R}^n \to \mathbb{R}$ be non-decreasing functions depending on a disjoint set of coordinates. Let $X = (X_1, \ldots, X_n)$ be a Gaussian random variable which is negatively correlated. Then we have*

$$E[f_1(X)f_2(X)] \leq E[f_1(X)]E[f_2(X)].$$

We now prove the following theorem.

▶ **Lemma 13.** *For any non-decreasing functions $f_1, f_2 : \mathbb{R}^n \to \mathbb{R}$ depending on disjoint set of coordinates that are 1-Lipschitz, we have for each $0 \leq t \leq T$,*

$$E[f_1(\mathbf{x}_t)f_2(\mathbf{x}_t)] \leq E[f_1(\mathbf{x}_t)]E[f_2(\mathbf{x}_t)] + 2n\sqrt{n}\varepsilon$$

Proof. We first ignore the errors introduced due to projection onto tight constraints and prove the inequality without the error term. We let $\mathbf{y}_t$ denote the corresponding fractional points. Thus $\mathbf{y}_t = \theta + \sum_{s=1}^{t-1} C(F_s)g_s$. We prove that

$$E[f_1(\mathbf{y}_t)f_2(\mathbf{y}_t)] \leq E[f_1(\mathbf{y}_t)]E[f_2(\mathbf{y}_t)]$$

by induction on t. For $t = 0$, $\mathbf{y}_0 = \theta$ and both sides are equal. Now, let the statement be true for $t \leq T-1$. We have $\mathbf{y}_{t+1} = \mathbf{y}_t + C_t g_t$ where we denote $C(F_t)$ by C_t. Thus we have

$$E[f_1(\mathbf{y}_{t+1}) f_2(\mathbf{y}_{t+1})|\mathbf{y}_t] = E[f_1(\mathbf{y}_t + C_t g_t) f_2(\mathbf{y}_t + C_t g_t)|\mathbf{y}_t]$$

But now observe that $C_t g_t$ is a Gaussian in $\mathbb{R}^n$ which is negatively correlated from Theorem 5. For any $\mathbf{y}_t$, we apply Theorem 12 to the functions $f_1(\mathbf{y}_t + \cdot)$ and $f_2(\mathbf{y}_t + \cdot)$, we obtain

$$\begin{aligned} E[f_1(\mathbf{y}_{t+1}) f_2(\mathbf{y}_{t+1})|\mathbf{y}_t] &= E[f_1(\mathbf{y}_t + C_t g_t) f_2(\mathbf{y}_t + C_t g_t)|\mathbf{y}_t] \\ &\leq E[f_1(\mathbf{y}_t + C_t g_t)|\mathbf{y}_t] E[f_2(\mathbf{y}_t + C_t g_t)|\mathbf{y}_t] \\ &= E[f_1(\mathbf{y}_{t+1})|\mathbf{y}_t] E[f_2(\mathbf{y}_{t+1})|\mathbf{y}_t]. \end{aligned}$$

Now taking expectations w.r.t. $\mathbf{y}_t$, we obtain that

$$E[f_1(\mathbf{y}_t) f_2(\mathbf{y}_t)] \leq E[f_1(\mathbf{y}_t)] E[f_2(\mathbf{y}_t)].$$

Now bound the distance between $\mathbf{x}_t$ and $\mathbf{y}_t$. We have

$$\|x_t - \mathbf{y}_t\|_2 = \|\sum_{i=1}^{t-1} \zeta_i\|_2 \leq \sqrt{n} \max_i \|\zeta_i\|_2 \leq n\sqrt{n}\varepsilon.$$

Since f_1 and f_2 are 1-Lipschitz, we have the result of the lemma. ◀

Acknowledgements. We thank Robin Pemantle for many discussions and comments.

References

1. Nima Anari, Shayan Oveis Gharan, and Alireza Rezaei. Monte carlo markov chain algorithms for sampling strongly rayleigh distributions and determinantal point processes. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, volume 49 of *JMLR Workshop and Conference Proceedings*, pages 103–115. JMLR.org, 2016. URL: `http://jmlr.org/proceedings/papers/v49/anari16.html`.
2. Arash Asadpour, Michel X. Goemans, Aleksander Madry, Shayan Oveis Gharan, and Amin Saberi. An o(log n/ log log n)-approximation algorithm for the asymmetric traveling salesman problem. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 379–389. SIAM, 2010. `doi:10.1137/1.9781611973075.32`.
3. Julius Borcea, Petter Brändén, and Thomas Liggett. Negative dependence and the geometry of polynomials. *Journal of the American Mathematical Society*, 22(2):521–567, 2009.
4. Petter Brändén and Rafael S González D'León. On the half-plane property and the tutte group of a matroid. *Journal of Combinatorial Theory, Series B*, 100(5):485–492, 2010.
5. Petter Brändén and Johan Jonasson. Negative dependence in sampling. *Scandinavian Journal of Statistics*, 39(4):830–838, 2012.
6. Chandra Chekuri, Jan Vondrak, and Rico Zenklusen. Dependent randomized rounding via exchange properties of combinatorial structures. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 575–584. IEEE, 2010.
7. Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Multi-budgeted matchings and matroid intersection via dependent rounding. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1080–1097. SIAM, 2011.

8 Devdatt P Dubhashi and Alessandro Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2009.

9 Devdatt P Dubhashi and Desh Ranjan. Balls and bins: A study in negative dependence. *BRICS Report Series*, 3(25), 1996.

10 Tomás Feder and Milena Mihail. Balanced matroids. In *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 26–38. ACM, 1992.

11 Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM (JACM)*, 53(3):324–360, 2006.

12 Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. A randomized rounding approach to the traveling salesman problem. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 550–559. IEEE, 2011.

13 Kumar Joag-Dev and Frank Proschan. Negative association of random variables with applications. *The Annals of Statistics*, pages 286–295, 1983.

14 Lap Chi Lau, Ramamoorthi Ravi, and Mohit Singh. *Iterative methods in combinatorial optimization*, volume 46. Cambridge University Press, 2011.

15 Shachar Lovett and Raghu Meka. Constructive discrepancy minimization by walking on the edges. *SIAM J. Comput.*, 44(5):1573–1582, 2015. doi:10.1137/130929400.

16 Ronald F Patterson, Wendy D Smith, Robert L Taylor, and Abolghassem Bozorgnia. Limit theorems for negatively dependent random variables. *Nonlinear Analysis: Theory, Methods & Applications*, 47(2):1283–1295, 2001.

17 Robin Pemantle. Towards a theory of negative dependence. *Journal of Mathematical Physics*, 41(3):1371–1390, 2000.

18 Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2002.

19 Mohit Singh and Nisheeth K. Vishnoi. Entropy, optimization and counting. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 50–59. ACM, 2014. doi:10.1145/2591796.2591803.

20 Aravind Srinivasan. Distributions on level-sets with applications to approximation algorithms. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 588–597. IEEE, 2001.

21 Ming Yuan, Chun Su, and Taizhong Hu. A central limit theorem for random fields of negatively associated processes. *Journal of Theoretical Probability*, 16(2):309–323, 2003.

Simultaneously Load Balancing for Every p-norm, With Reassignments*

Aaron Bernstein[1], Tsvi Kopelowitz[2], Seth Pettie[3], Ely Porat[4], and Clifford Stein[5]

1 **Columbia University, New York, USA**
bernstei@gmail.com
2 **University of Michigan, Ann Arbor, USA**
kopelot@gmail.com
3 **University of Michigan, Ann Arbor, USA**
pettie@umich.edu
4 **Bar Ilan University, Ramat Gan, Israel**
porately@cs.biu.ac.il
5 **Columbia University, New York, USA**
cliff@ieor.columbia.edu

Abstract

This paper investigates the task of load balancing where the objective function is to minimize the p-norm of loads, for $p \geq 1$, in both static and incremental settings. We consider two closely related load balancing problems. In the bipartite matching problem we are given a bipartite graph $G = (C \cup S, E)$ and the goal is to assign each client $c \in C$ to a server $s \in S$ so that the p-norm of assignment loads on S is minimized. In the graph orientation problem the goal is to orient (direct) the edges of a given undirected graph while minimizing the p-norm of the out-degrees. The graph orientation problem is a special case of the bipartite matching problem, but less complex, which leads to simpler algorithms.

For the graph orientation problem we show that the celebrated Chiba-Nishizeki peeling algorithm provides a simple linear time load balancing scheme whose output is an orientation that is 2-competitive, in a p-norm sense, for all $p \geq 1$. For the bipartite matching problem we first provide an offline algorithm that computes an optimal assignment. We then extend this solution to the online bipartite matching problem with reassignments, where vertices from C arrive in an online fashion together with their corresponding edges, and we are allowed to reassign an amortized $O(1)$ vertices from C each time a new vertex arrives. In this online scenario we show how to maintain a single assignment that is 8-competitive, in a p-norm sense, for all $p \geq 1$.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Online Matching, Graph Orientation, Minimizing the p-norm

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.51

1 Introduction

Online algorithms for load balancing have received much attention in recent years. There are many variants of load balancing, and in this paper we consider two closely related load balancing problems – bipartite matchings and graph orientations. These problems have many natural applications in areas such as internet advertising, social networks, and routing. In the standard online paradigm, an object such as a vertex or edge arrives online

* Research supported in part by NSF grants CCF-1421161, CCF-1514383, and CCF-1637546.

© Aaron Bernstein, Tsvi Kopelowitz, Seth Pettie, Ely Porat, and Clifford Stein;
licensed under Creative Commons License CC-BY

and then a decision, such as orienting an edge or matching two vertices, must be made immediately. This decision is *irrevocable*. It is well understood that, for most problems, an online algorithm cannot perform as well as an offline algorithm which knows the future requests. Sometimes the gap can be quite large.

It is therefore natural to consider giving the online algorithm a small amount of additional power. This idea has taken many forms such as resource augmentation, randomization, advice, or knowledge of an input distribution. Here we study the idea of allowing the online algorithm to "redo" some of its earlier decisions. Since we are studying polynomial-time solvable problems such as matching and orientation, we have to be sure not to allow too many opportunities to redo decisions, as we would then just simulate an offline algorithm. In particular, we will allow edge reorientations or reassignments, but will focus on algorithms that allow only a small number of reassignments. Our results will, in general, show that by allowing a small number of reassignments, we can maintain a much better matching or orientation.

The p-norm

Typically one evaluates a matching by its size or total cost and one evaluates an orientation by its maximum edge out-degree, or load. In this paper, we consider a more general class of objective functions, evaluating the p-norm, for any $p \geq 1$. For a vector $X = (x_1, x_2, \ldots, c_k)$ the p-norm of X is denoted by $\|X\|_p = (\sum_{i=1}^{k} x_i^p)^{1/p}$. Often, with a load-balancing problem, there is a tension between finding a solution that minimizes the maximum load and one that fairly distributes load among various agents. Considering only the maximum load ignores fairness among the non-maximum load nodes and considering only the average ignores the possibility of a solution with one or more nodes of very high load. This tradeoff has been studied in various ways in previous work, e.g. [19, 24, 14, 25, 4]. We address this tradeoff by considering p-norms. By using a small constant $p > 1$, it is well understood that one balances these two competing objectives. In this paper, we will give algorithms that perform well on all norms simultaneously. This not only shows that the algorithm does not need to be specialized to the norm, but also shows that there exist solutions that perform well in all norms simultaneously.

1.1 Edge orientation and bipartite matching

We investigate the goal of minimizing the p-norm of loads by considering two closely related problems.

Graph orientations

In the graph orientation problem, the goal is to orient (direct) the edges of an undirected graph G while minimizing some function of the out-degrees, and in our case the p-norm. If we consider an *out-degree vector* X of length n where x_i is the out-degree of vertex v_i, then one can word this goal as minimizing $\|X\|_p$.

Bipartite matching

In the online bipartite matching with reassignments problem a bipartite graph contains two sets of vertices: there is a set of vertices S (*server* vertices) that is fixed in advance, and a set of vertices C (*client* vertices) that grows over time. In particular, at each time step, a new client c is inserted into C along with edges from c to some non-empty subset of S. We

say that a server $s \in S$ is a *neighbor* of c if edge (c, s) exists. Given any time t, we let C_t be the set of client vertices at time t. We always let C refer to the set of vertices at the *current* time t. Similarly, we let $G_t = (V_t, E_t)$ refer to the resulting bipartite graph at time t (so $V_t = C_t \bigcup S$), and G refer to the graph at the current time t.

The online algorithm must at all times maintain a *global assignment* function $A : C \to S$, which assigns each client $c \in C$ to some server $s \in S$. Whenever a new client c is inserted into C, it must immediately be *assigned* to one of its neighbors in S. However, unlike in standard online matching, we also allow the algorithm to reassign $O(1)$ vertices from C each time a client arrives. Given any $s \in S$, let $\text{LOAD}_A(s) = |\{c \in C : A(c) = s\}|$ refer to the number of client vertices assigned to s by A. Let $\overline{\text{LOAD}}_A = (\text{LOAD}_A(s_1), \text{LOAD}_A(s_2), \ldots, \text{LOAD}_A(s_k))$ where $S = \{s_1, s_2, \ldots, s_k\}$. The vector $\overline{\text{LOAD}}_A$ is is called the *load vector of* A. The goal is to maintain an assignment A that minimizes $\|\overline{\text{LOAD}}_A\|_p$.

Notice that if each client arrives with exactly two neighboring vertices in S, then the online bipartite matching problem corresponds to an incremental version of the graph orientation problem (where edges arrive online and orientations are allowed to be flipped) by considering S as the set of vertices and C as the set of edges.

1.2 Our results

Our main result is to show that for the online bipartite matching with reassignments problem we can maintain an 8-approximation to the optimal assignment, while reassigning a small number of clients. To do so, we first show how to find an optimal offline solution using an adaptation of augmenting paths as follows. Given a global assignment A we orient[1] the edges of G as follows: an edge (s, c) is oriented $s \to c$ if $A(c) = s$, and otherwise it is oriented $c \to s$. This oriented graph is denoted by G_A.

▶ **Definition 1.** For a bipartite graph $G = (C \cup S, E)$ and a global assignment A, a directed path P in G_A from server $x \in S$ to server $y \in S$ is an augmenting path if $\text{LOAD}_A(y) \leq \text{LOAD}_A(x) - 2$.

▶ **Theorem 2.** *For a bipartite graph $G = (S \cup C, E)$ and for any $1 \leq p < \infty$, a global assignment A is optimal with respect to the p-norm if and only if G_A has no augmenting paths.*

Our online algorithm will always keep track of the optimal offline solution in the background (Theorem 2 provides a simple algorithm for keeping track of OPT as new clients arrive), and use it to influence its online assignments. Note that even though the online algorithm knows OPT it does not directly use the same assignment as OPT because maintaining OPT as new vertices are inserted into C may require too many reassignments.

Let OPT_t be an optimal assignment at time t. We are now ready to state our main theorem for online assignment.

▶ **Theorem 3.** *There exists an algorithm for online assignment which only performs an amortized $O(1)$ reassignments per time step (i.e. $O(1)$ per new client vertex), and ensures that at every time t, for any $s \in S$, $\text{LOAD}_A(s) \leq 8 \cdot \text{LOAD}_{\text{OPT}_t}(s)$. Note that the resulting assignment is always 8-competitive to the optimal assignment for any p-norm where $p \geq 1$, including $p = \infty$.*

[1] Notice that this is not the same orientation as in the graph orientation problem, although the two are related.

Results for graph orientation

While Theorem 2 provides a method for finding an optimal orientation in G in polynomial time (since the graph orientation problem is a special case), the runtime (via a straightforward implementation) is $O(m^2)$ where m is the number of edges. Instead, based on the Chiba and Nishizeki [7] triangle enumeration algorithm, we provide a simple linear time algorithm for finding an orientation that is a 2-approximation to the optimal p-norm. For an orientation $\mathcal{O}$ and a vertex u Let $d^+_{\mathcal{O}}(u)$ be the out-degree of u in the graph oriented by $\mathcal{O}$.

▶ **Theorem 4.** *There exists a linear time algorithm for the graph orientation problem on a graph $G = (V, E)$ producing an orientation $\mathcal{O}$ such that for any $p \geq 1$ and any vertex $u \in V$, $d^+_{\mathcal{O}}(u) \leq 2 \cdot d^+_{\text{OPT}}(u)$ where* OPT *is an optimal orientation with respect to the p-norm.*

We emphasize that one can use Theorem 3 to solve an incremental/online version of the graph orientation problem with a constant number of edge flips to maintain an orientation with a close to optimal p-norm of the out-degrees. The nice benefit of the proof of Theorem 4 is that it introduces and investigates a new natural combinatorial measurement of graphs, called the *local degeneracy*, which may be of independent interest.

1.3 Overview and Techniques for Bipartite Matching

To prove Theorem 3 we start by proving a simpler theorem.

▶ **Theorem 5.** *Say that for some specific time t_f we know in advance an upper bound on $\text{LOAD}_{\text{OPT}_{t_f}}(s)$ for every $s \in S$: that is, we know some function $u : S \to \mathbb{N}$ such that $\text{LOAD}_{\text{OPT}_{t_f}}(s) \leq u(s)$ for every $s \in S$. Then, there is an online assignment protocol that performs an amortized $O(1)$ reassignments per time step, and guarantees that at time t_f, every server $s \in S$ has $\text{LOAD}_{t_f}(s) \leq 2u(s)$.*

The proof of Theorem 5 is based on a recent result of Gupta, Kumar, and Stein [16] that also studies the problem of online assignment with reassignments, but they solve a simpler problem of trying to minimize the *maximum* load (i.e. the L_∞ norm of the loads) while performing as few reassignments as possible. In particular, they achieve the following results (the second result is a combination of the first result and the standard guess and double technique):

▶ **Theorem 6.** *[16] Say that we know in advance that at some final time step t_f, there is a solution in which every server $s \in S$ has load at most k. Then, there is an online assignment protocol that performs an amortized $O(1)$ reassignments per time step, and achieves a maximum load of $2k$ at time t_f.*

▶ **Theorem 7.** *[16] There exists an algorithm for online assignment which performs $O(1)$ reassignments per time step and has the following guarantee: at every time step t, if there exists an offline solution in which every $s \in S$ has a load of at most k_t, then the online protocol achieves a maximum load of at most $8k_t$.*

Although there is a trivial extension from Theorem 6 to Theorem 5, there is no trivial extension from Theorem 7 to Theorem 3. The intuition for why not knowing the optimal maximum load in advance poses a problem is as follows. Loosely speaking, the analysis of Theorem 6 (where the optimal max load k is known in advance) relies on the fact that the server vertices are only getting more and more constrained: if we know in advance that the

final load should be k, then since the load on every server $s \in S$ can only increase over time, as more clients arrive less and less capacity remains before we reach load k. But if the loads are not known in advance, then the algorithm must occasionally increase the desired load on s from k to say $2k$, which effectively makes s *less* constrained, and destroys the analysis in Theorem 6.

Theorem 7 is able to very easily overcome this obstacle because as long as we only care about the maximum load, the desired load for all $s \in S$ changes at the same time: as more vertices are inserted into C, the algorithm sometimes realizes that max load k is no longer feasible, so it increases the max allowable load to $2k$ for all vertices $s \in S$. Such global changes are easy to handle by effectively restarting the entire algorithm every time the desired max load doubles. But this approach does not work for Theorem 3 because here the algorithm must aim for a different load on every server, and these loads will change at different times. This prevents us from applying a global restart, and locally changing the allowable load on one server ends up destroying the analysis for the not-yet-changed parts of the graph.

We show, however, that it is nonetheless possible to do a reduction from Theorem 3 (loads not known in advance) to Theorem 5 (loads known in advance), but the reduction is significantly more complicated, and requires a more careful analysis of how the optimal offline assignment OPT relates to our online assignment. The main idea is to create another graph G^* by carefully duplicating servers with exponentially increasing capacities. Once the first i servers are full we begin using the $i+1$'th server. We show that maintaining an optimal solution in G^* can be used to maintain a close to optimal solution in G, and then we show how to maintain a close-to-optimal solution in G^*.

1.4 Overview and Techniques for Graph orientations: The Local Degeneracy

When considering the ∞-norm of out-degrees, a natural parameter to consider is the *arboricity* or *degeneracy* of the graph. The degeneracy of an undirected graph $G = (V, E)$ is the largest minimum degree of any subgraph of G. Formally the degeneracy of G is $\delta(G) = \max_{U \subseteq V} \min_{u \in U} d_{G_U}(u)$ where d_{G_U} is the degree of u in the subgraph of G induced by U. The degeneracy is roughly equal (up to constant factors) to other well known graph parameters such as the arboricity, strength, or thickness of a graph. Of particular interest is the arboricity $\alpha(G)$ of G since the maximum out-degree of any orientation is at least $\alpha(G) - 1$, and $\alpha(G) \leq \delta(G) \leq 2\alpha(G)$-1.

Chiba and Nishizeki [7] gave a linear time algorithm, called the *CN-peeling* algorithm, that computes a $\delta(G)$-orientation as follows. Repeatedly remove a vertex u in (the current) G with smallest (current) degree, together with the edges E_u touching u at the time u is removed. For each u, all of the edges E_u are oriented as leaving u. The out-degree of each vertex is shown to be at most $\delta(G)$ using straightforward arguments. The CN-peeling algorithm provides an acyclic orientation of G, where for each vertex u all of the edges E_u are oriented as leaving u. This acyclic orientation is used to enumerate all triangles of the graph in time $O(m \cdot \delta(G))$ with the following observation: for each triangle v_1, v_2, v_3 in G there must be at least one vertex of the three with both edges oriented outwards. Thus, it suffices to consider, for each vertex, all pairs of outgoing edges.

Local versus global parameters

While the algorithm of [7] is essentially tight for worst-case inputs (see [21, 22]), for many graphs the runtime bound of $O(m \cdot \delta(G))$ is not tight. For example, consider a graph G composed of a clique on $n^{1/3}$ vertices, and a tree on the rest of the vertices, with the tree being connected to the clique with one edge. The degeneracy of G is $\Theta(n^{1/3})$ and so $O(m \cdot \delta(G)) = O(n^{4/3})$. However, it is obvious that triangle enumeration through the CN-peeling algorithm runs in $O(n)$ time.

Why is this analysis not tight enough? Because the degeneracy is a global parameter of G which is largely due to a small part of G being dense. However, in the analysis, this global parameter should not affect the time cost of other sparser parts in G. It would therefore be more beneficial to express the runtime using tighter graph parameters which are more appropriate in determining the local cost of each vertex.

Local degeneracy

We extend the notion of degeneracy by introducing a new local sparseness measurement. The *local degeneracy* (LD) of a vertex v in G is $\ell_v = \max_{U \subseteq V, v \in U} \min_{u \in U} d_{G_U}(u)$. In words, the LD of v is the largest minimum degree of any subgraph of G that contains v. It is straightforward to show that in the orientation obtained from the CN-peeling algorithm the out-degree d_u^+ of a vertex u is at most its LD ℓ_u. The runtime of the CN-peeling algorithm for triangle enumeration is then $\sum_{v \in V} (d_u^+)^2 \leq \sum_{v \in V} (\ell_u)^2$, which is always at most $O(m \cdot \delta(G))$, but could be much less. In the example graph G above, this turns out to be $O(n)$. So, at least in the example, LD captures a locality notion that better expresses the runtime of the peeling algorithm.

We prove that LD captures a fundamental property of graph orientations stated in the following theorem.

▶ **Theorem 8.** *For an undirected graph $G = (V, E)$ and for any $1 < p < 0$, let* OPT *be any optimal orientation of the edges with respect to the p-norm. Let $d^+_{\text{OPT}}(v)$ be the out-degree of $v \in V$ when G is oriented according to* OPT*. Then $d^+_{\text{OPT}}(v) > \ell_v/2 - 1$. For the ∞-norm, the same is true for some optimal orientation.*

Theorem 8 has two important consequences. The first is that the CN-peeling algorithm gives an orientation whose out-degrees are 2-competitive to the optimal orientation with respect to the p-norm (Theorem 4). The second implication is that any algorithm that uses an acyclic orientation for enumerating all triangles in a given graph, by separately considering for each vertex all pairs of its outgoing edges, cannot be asymptotically faster than the CN-peeling algorithm.

Local degeneracy and the k-core

The *k-core* of G is the maximum subgraph of G with minimum degree at least k, and is denoted by G_k. As it turns out, if $u \in G_k - G_{k+1}$ then $\ell_u = k$ (more on this in Section 4). While the notion of a k-core is certainly not new, the focus from an algorithmic perspective has mainly been on computing the k-core of a graph for a given k. In other words, the focus has been on the k-core from a global prospective of the graph. Our focus is on a local property of vertices, which is captured by the local degeneracy of each vertex. Thus, we find the terminology of *local degeneracy* to be more appropriate for our objectives.

1.5 Related work

There is a large body of work that deals with the tradeoff between reassignment and the quality of an online solution. E.g., consider the problem of load balancing, where each arriving task has a processing time but can run only on some subset of the servers. Here, Phillips and Westbrook [28] and Westbrook [33] showed that a small number of reassignments (linear in the number of tasks) improve the quality of the solution, getting a constant competetive ratio, compares to logarithmic lower bounds in the case without reassignments [3]. There has also been work on the case of identical machines [30, 13, 29, 32, 12] and reassignments for other online problems such as Steiner Tree [18, 26, 15].

Related work on online bipartite matching

For online matching, the maximization version has been studied extensively. In this model all arriving nodes do not need to be matched, and the goal is to maximize the reward accrued by successfully matching a large number/weight of these nodes. It is well known that the greedy algorithm is 1/2-competitive for maximum matching. There are many papers on variants including when the capacity on one side is large relative to the requests, or when there is stochastic information, or when randomization is allowed. These results are orthogonal to our line of investigation, and we omit a detailed discussion.

Most relevant to our work is a paper by Gupta, Kumar and Stein [16] that studies the problem of online assignment in the same model as this paper. For online bipartite matching, where the left vertices arrive online and must be matched to the right vertices, they give an algorithm that reassigns the left vertices an (amortized) constant number of times, and maintains a constant factor to the optimal load on the right vertices. They then extend this result in several ways. For restricted machine scheduling with arbitrary sized jobs, they give an algorithm that maintains load which is $O(\log \log mn)$ times the optimum, and reassigns each job only an (amortized) constant number of times. They also give an algorithm for online flow that reroutes flow an (amortized) constant number of times while achieving constant factor approximation to the congestion. The bounds in the SODA proceedings version of [16] are correct, but there is an error in the proof. Work on this paper revealed that proof, which the authors have since corrected (the corrected verion has not yet been made public). Our work generalizes the results in that paper.

Related work on graph orientations

The task of maintaining an edge orientation with the goal of minimizing the maximum out-degree has also received a lot of attention [5, 20, 17]. Edge orientations have many algorithmic applications including "color-coding" [1], adjacency queries [8, 5, 23, 20], short-path queries [24], load balancing [6], maximal matchings [27], pattern matching [2], counting subgraphs in sparse graphs [9], prize-collecting TSPs and Steiner Trees [10], reporting all maximal independent sets [11], answering dominance queries [11], subgraph listing problems (listing triangles and 4-cliques) in planar graphs [8], and computing the girth [24].

2 Optimal Offline Bipartite Matching

Proof of Theorem 2. If A is optimal with respect to the p-norm then clearly there are no augmenting paths since flipping an augmenting path leads to a new assignment with smaller p-norm, due to convexity. So we focus on proving that if there are no augmenting paths in

G_A then A must be optimal for any p-norm with $1 < p < \infty$. For the following let p be any p in the range.

Let OPT_p be the optimal global assignment with respect to p. We partition the set of vertices in S as follows. $S^< = \{u \in S | \text{LOAD}_A(u) < \text{LOAD}_{\text{OPT}_p}(u)\}, S^> = \{u \in S | \text{LOAD}_A(u) > \text{LOAD}_{\text{OPT}_p}(u)\}$, and $S^= = S - \{S^< \cup S^>\}$. Let $E^* = \{(c,s) \in E | A(c) = s \in A \wedge \text{OPT}_p(c) \neq s\}$ be the set of edges in the symmetric difference between G_A and G_{OPT_p}. Let A^* by a global assignment of $G^* = (S \cup C, E^*)$ where each assignment of a vertex from C is made according to its assignment in A. Notice that each vertex in C has either degree 0 or 2 in G^*.

Let C be a directed cycle in $G^*_{A^*}$. Notice that if we were to flip all of the edges of C in G_A, then the out-degree of each vertex would not change due to the flip, and so the p-norm of the corresponding assignment would remain the same. However, after flipping the edges in C these edges are oriented in the same direction in both G_A and G_{OPT_p}. So we may assume without loss of generality that $G^*_{A^*}$ contains no cycles, since otherwise we iteratively pick a cycle and flip it until no cycles are left.

Given that $G^*_{A^*}$ does not contain any directed cycles, then either E^* is empty, in which case we are done, or $G^*_{A^*}$ is a directed acyclic graph with some edges. We say a directed path from $u \in S$ to $v \in S$ in $G^*_{A^*}$ with length at least 1 is a maximal path if there is no edge entering u and no edge leaving v. Notice that a maximal path cannot start in S^- since every vertex in S^- has edges leaving it, and it cannot start in $S^=$ since every vertex in $S^=$ has the same number of incoming and outgoing edges. Similarly, a maximal path cannot end in S^+ or $S^=$. Thus, all maximal paths must begin in S^+ and end in S^-.

Consider a maximal path P from $u \in S^+$ to $v \in S^-$. We will prove that $\text{LOAD}_A(u) = \text{LOAD}_A(v) + 1$. From this it will follow that we can flip P in G_A thereby swapping the loads of u and v in A. So the p-norm of the loads of this new assignment is the same as the $\|\text{LOAD}_A\|_p$. We can then iteratively flip maximal paths until we obtain the assignment OPT_p, implying that $\|\text{LOAD}_A\|_p = \|\text{LOAD}_{\text{OPT}_p}\|_p$ as required.

If $\text{LOAD}_A(u) > \text{LOAD}_A(v) + 1$ then P is an augmenting path in G_A, which contradicts the assumption. If $\text{LOAD}_A(u) < \text{LOAD}_A(v) + 1$, then since $u \in S^+$ and $v \in S^-$ we have that $\text{LOAD}_{\text{OPT}_p}(u) \leq \text{LOAD}_A(u) + 1 \leq \text{LOAD}_A(v) + 1 \leq \text{LOAD}_{\text{OPT}_p}(v) + 2$. This implies that the reverse of P, which is a path in G_{OPT_p}, is an augmenting path, contradicting OPT_p being an optimal orientation with respect to the p-norm. ◀

By Theorem 2 an orientation that is optimal with respect to some $1 < p < \infty$ is optimal with respect to any such p, and so we denote any such orientation by OPT. We also call OPT the optimal orientation for G.

▶ **Corollary 9.** *If an orientation has no augmenting paths, then it is optimal for the ∞-norm.*

3 Online Matching with Reassignments: Preliminaries and Main Theorem Statement

Proof of 5. The proof shows how to reduce this problem to the one in Theorem 6. In particular, recall that $G = (V, E)$ always corresponds to the graph induced by $V = C \cup S$ at the current time t (so G changes as new clients are inserted). Instead of directly maintaining an online assignment in G, we create a slightly different graph $G' = (V', E')$ which is essentially equivalent to G, but where we only need to concern ourselves with the ∞-norm. We define $V' = S' \cup C$, where the set S' contains $u(s)$ copies of every vertex $s \in S$, labeled $s_1, s_2, ..., s_{u(s)}$. Then, for every edge $(c, s) \in E$, we add edges $(c, s_1), (c, s_2), ..., (c, s_{u(s)})$ to E'.

It is clear that since $\text{LOAD}_{\text{OPT}_{t_f}}(s) \leq u(s)$, at time t_f there is an assignment in G' in which every vertex $s_i \in S$ has load at most 1. But this means via applying Theorem 6 that there is an assignment protocol for G' that only requires an amortized $O(1)$ reassignments per time step, and that maintains an assignment in G' where at time t_f, every vertex s_i has load at most 2. This assignment for G' then translates directly to the desired assignment in G: assigning a client c to some copy $s_i \in V'$, corresponds to assigning c to $s \in V$. We only perform a reassignment in G when we perform one in G' (but not necessarily vice versa: a reassignment between two copies s_i and s_j in G' does not lead to a reassignment in G), so the number of reassignments in G will also be at most amortized $O(1)$ per time step. Since every copy $s_i \in V'$ has at most two client vertices assigned to it, and there are $u(s)$ copies of s in V', we will end up with $\text{LOAD}_{t_f}(s) \leq 2u(s)$, as desired. ◀

Defining a Capacity Function

Unlike Theorem 5, Theorem 3 does not specify a particular finish time t_f, so the online assignment must be a good approximation to OPT at *all* time steps. Thus, in order to rely on Theorem 5, we need to define a *fixed* capacity function $u : S \to \mathbb{N}$ such that at *all* times t we have that for every $s \in S$, $\text{LOAD}_{\text{OPT}_t}(s) \leq u(s)$.

This seems contradictory – how can u be an upper bound on $\text{LOAD}_{\text{OPT}_t}$ if the first is fixed and the latter increases with t – but the idea is to capture changes in $\text{LOAD}_{\text{OPT}_t}(s)$ by creating many copies of each $s \in S$, each with different capacities $u(s)$. (Note: these copies are unrelated to the copies used in the proof of Theorem 5.)

Recall that $G = (V, E)$ refers to the current version of the graph. We define another graph $G^* = (V^*, E^*)$ that also changes as new clients are inserted. We start by defining a new set of servers S^*, which contains an infinite number of copies for each vertex $s \in S$, labeled: $s_0, s_1, s_2, s_3...$, where copy s_i will have capacity $u(s_i) = 2^{i+1}$. Our algorithm will not have to actually handle an infinite number of copies because originally all the copies s_i of a vertex $s \in S$ are *closed*: there are no edges from C to a closed copy s_i, so s_i will never have any client vertices assigned to it and can be entirely ignored.

Recall that as new clients are inserted into the graph, we are always keeping track of an optimal offline solution OPT, and that by the proof of Theorem 2 $\text{LOAD}_{\text{OPT}}(s)$ never decreases for any $s \in S$. Let OPT_t be an optimal solution at time t. When a new vertex $c \in C$ arrives at time $t(c)$, we do the following to the graph G^*:

- If for some vertex $s \in S$, we have $\text{LOAD}_{\text{OPT}t(c)-1}(s) = 2^i - 1$ and $\text{LOAD}_{\text{OPT}t(c)}(s) = 2^i$, then we open copy $s_i \in S^*$.
- For every edge (c, s) in G, we add an edge (c, s_i) to E^* for every *open* copy s_i.

Note that our end goal is to apply Theorem 5 to G^*, so it is crucial that G^* evolves according to the definition of an online assignment problem given in Section 3: at each time step, a client arrives with all of its incoming edges, and no new edges are ever added or removed. In particular, when a new copy s_i opens up at time t, we do NOT add edges from (c, s_i) to G^* for client vertices $c \in C$ that arrived before time t. However, the lack of edges from opened server copies in S^* to older client vertices in C leads to the problem that G^* ends up being quite different from the main graph G. In particular, we can in theory imagine a situation where OPT assigns many client vertices $c_1, c_2, ..., c_q$ to some vertex $s \in S$, and yet in G^* we cannot assign all these c_i to copies of s in S^*, because even though there are many open copies of s (because $\text{LOAD}_{\text{OPT}}(s) = q$ is large), it might be the case that most of the copies s_i were created *after* the clients $c_1, ..., c_q$ arrived, so there no edges from these c_i to the copies of s in S^*. The following lemma shows that G^* is nonetheless a good

approximation to G: we first apply the Lemma to Theorem 3, and then proceed to prove the lemma.

▶ **Lemma 10.** *At all times t, there exists an assignment in G_t^* of vertices $c \in C$ to vertices $s_i \in S^*$ such that $\text{LOAD}(s_i) \leq 2^{i+1} = u(s_i)$.*

Proof of Theorem 3. The proof relies on using Theorem 5 to maintain an online assignment in G^*. Note that we do not need to specify a specific time t_f as in Theorem 5, because we know from Lemma 10 that at ALL times t there is an assignment from C to S^* in which every $s_i \in S^*$ has load at most $u(s_i) = 2^{i+1}$. Thus, by Theorem 5 we can maintain an assignment from C to S^* that only uses an amortized $O(1)$ reassignments per new client, and in which for all times t we have $\text{LOAD}(s_i) \leq 2u(s_i) = 2^{i+2}$.

The assignment from C to S^* in G^* then translates directly into an assignment from C to S in G: for every assignment $c \to s_i$, if $s_i \in S^*$ is a copy of $s \in S$, then we assign $c \to s$. The number of reassignments in G is also amortized $O(1)$ per new client, since every reassignment in G corresponds directly to one in G^* (the opposite is not necessarily true: a reassignment in G^* between two copies s_i and s_j does not translate to a reassignment in G). All we have left to show is that for each $s \in S$, and for all times t, we always have $\text{LOAD}_t(s) \leq 8\text{LOAD}_{\text{OPT}_t}(s)$. To see this, note that $\text{LOAD}_t(s) = \sum_i \text{LOAD}_t(s_i)$, where we know that $\text{LOAD}_t(s_i) \leq 2^{i+2}$ if copy i is open, and $\text{LOAD}_t(s_i) = 0$ if copy i is closed. But we only open copy s_i if $\text{LOAD}_{\text{OPT}_t}(s) \geq 2^i$, so letting j be an index for which $2^j \leq \text{LOAD}_{\text{OPT}_t}(s) < 2^{j+1}$, we have that

$$\text{LOAD}_t(s) = \sum_i \text{LOAD}_t(s_i) = \sum_{i \leq j} \text{LOAD}_t(s_i) \leq \sum_{i \leq j} 2^{i+2} < 2^{j+3} \leq 8\text{LOAD}_{\text{OPT}_t}(s). \qquad \blacktriangleleft$$

We now turn to the proof of Lemma 10, which requires a more careful analysis of the offline solution OPT in G.

▶ **Definition 11.** If at some time t OPT assigns $c \in C$ to $s \in S$ – whether because c just arrived, or because OPT chooses to reassign c at time t – we say that the assignment $c \to s$ has *level* k if $\text{LOAD}_{\text{OPT}_t}(s) = k$.

Note that if we look at all the different assignments of some $c \in C$, their level is monotonically increasing over time. This is because OPT always makes the lowest level assignment possible and LOAD_{OPT} is monotonically increasing. So if at time t_1 there was an assignment $c \to s_1$ of level k_1, and at time $t_2 > t_1$ there was an assignment $c \to s_2$ of level $k_2 < k_1$, then OPT would have instead assigned c to s_2 at time t_1.

▶ **Definition 12.** We say that a vertex $c \in C$ has *initial level* k if the assignment $c \to s$ performed by OPT when c first arrives is a level k assignment. Finally, we say that a vertex $s \in S$ is the *final level k server* of some $c \in C$ if the assignment $c \to s$ is the last level k assignment OPT performs on c.

▶ **Claim 13.** *Say that vertex $s \in S$ is the final level k server of vertex $c \in C$. Then, if at time t OPT reassigns c from s to s', we must have that $\text{LOAD}_{\text{OPT}_t}(s) > k$.*

Proof. OPT only has to reassign c from s to s' if it assigned some other c' to s. Now, we know that the assignment $c \to s'$ could not have been of level less than k because earlier c had a level k assignment to s, and the level of assignments of c is monotonically increasing. We also know that the reassignment $c \to s'$ cannot be of level k since s was the *final* level k server of c. Thus, the assignment $c \to s'$ is of level at least $k+1$, so by definition $\text{LOAD}_{\text{OPT}_t}(s') \geq k+1$. But this means that $\text{LOAD}_{\text{OPT}_t}(s) \geq k+1$, because otherwise OPT would not have reassigned c from s to s'. $\blacktriangleleft$

▶ **Claim 14.** *Let $s \in S$ be the final level k server for vertices $c_1, c_2, c_3, ..., c_q$; then $q \leq k$.*

Proof. Say, for contradiction, that $q \geq k+1$. Let t_i, for $1 \leq i \leq q$, be the time at which s became the final level k sever for c_i. Without loss of generality, let $t_1 < t_2 < ... < t_q$. We want to argue that at the end of time t_q we have $\text{LOAD}_{\text{OPT}t_q}(s) \geq k+1$, which contradicts s being the final k server for c_q. There are two cases to consider. The first case is that OPT still assigns all of $c_1, c_2, ..., c_{q-1}$ to s at the end of t_q; then, since OPT also assigns c_q to s at time t_q, we have that $\text{LOAD}_{\text{OPT}t_q}(s) \geq q \geq k+1$. The second case is that by time t_q some c_i is no longer assigned to s in OPT, in which case by Claim 13, we must have that $\text{LOAD}_{\text{OPT}t_q}(s) \geq k+1$. ◀

Proof of Lemma 10. We say that a vertex $c \in C$ has *priority* i if the initial level of c is in $(2^i, 2^{i+1}]$. We now argue the following: if vertex $c \in C$ has priority i, then for every edge $(c, s) \in E$, we have that E^* must contain edges $(c, s_0), (c, s_1), ...(c, s_i)$ (E^* might also contain edges from c to later copies of s). Let us say for the sake of contradiction that edge $(c, s_i) \notin E^*$. This can only happen if s_i was not yet open at time $t(c)$, where $t(c)$ is the arrival time of c, and so in particular if $\text{LOAD}_{\text{OPT}t(c)}(s) < 2^i$. Say that at time $t(c)$, c was assigned to $s' \in S$. Since c has priority i, we know that at the end of $t(c)$ we have $\text{LOAD}_{\text{OPT}t(c)}(s') > 2^i$. This inequality yields the desired contradiction, since OPT is not optimal, as it would have been better off assigning c to s instead of s' at time $t(c)$.

Let $C_t^i \subseteq C_t$ contain all vertices in C_t that have priority exactly i. We first observe that there is an assignment in G from all vertices $c \in C_t^i$ to vertices in S in which every $s \in S$ has load at most 2^{i+1}. In particular, we can simply assign each $c \in C_t^i$ to its final level 2^{i+1} server, and by claim 14, each $s \in S$ will end up with at most 2^{i+1} client vertices assigned to it. This implies that at all times t, there exists an assignment in G^* that assigns each vertex $c \in C_t^i$ to some ith copy $s_i \in S^*$, while maintaining $\text{LOAD}(s_i) \leq 2^{i+1} = u(s_i)$ (and $\text{LOAD}(s_j) = 0$ for all $j \neq i$.) The reason is that we can take the above assignment from C_t^i to S in the main graph G, and for every assignment $c \to s$ in G, we can simply assign $c \to s_i$ in G^*; we know from the above paragraph that edge (c, s_i) necessarily exists in G^* because c has priority i. This completes the proof of Lemma 10, since our final assignment in G^* from C_t to S^* is simply the union among all priorities i of all the assignments from C_t^i to the ith copies $s_i \in S^*$. ◀

4 Graph Orientation

Orienting with local degeneracy

We use Chiba and Nishizeki's peeling algorithm. Let $(v_1, \ldots, v_n)$ be the order of vertices as encountered by the peeling algorithm. Recall that E_u is the set of edges touching u at the time u is peeled.

▶ **Lemma 15.** $|E_{v_i}| \leq \ell_{v_i}$.

Proof. The subgraph of G at the point where v_i is peeled has a minimum degree of $|E_{v_i}|$ so ℓ_{v_i} must be at least $|E_{v_i}|$. ◀

Consider the k-core of G, denoted by G_k. Clearly, every vertex in G_k must have local degeneracy at least k. Seidman in [31] showed that by recursively deleting (peeling) vertices with degree less than k one obtains G_k.

▶ **Lemma 16.** *For v_i let $j \leq i$ be the smallest integer such that $|E_{v_j}| = \max_{1 \leq k \leq i}\{|E_{v_k}|\}$. Then $\ell_{v_i} = |E_{v_j}|$.*

Proof. When v_j is peeled, all vertices must have degree at least $|E_{v_j}|$ and so $\ell_{v_i} \geq |E_{v_j}|$. Furthermore, by the correctness of the process described in Seidman in [31], right before v_j is peeled the remaining graph must be a $|E_{v_j}|$-core of G, and if $\ell_{v_i} > |E_{v_j}|$ then the algorithm must encounter a vertex $v_{j'}$ where $j < j' \leq i$ such that $\ell_{v_i} = |E_{v_{j'}}| > |E_{v_j}|$ contradicting the definition of j. ◀

Proof of Theorem 4. The whole proof relies on the following simple claim: for $v, w \in V$, if there is a directed path in G oriented by OPT from w to v, then $d^+_{\mathrm{OPT}}(w) \leq d^+_{\mathrm{OPT}}(v) + 1$. This is because otherwise, OPT could be converted to a better solution by flipping the path from w to v.

Assume by contradiction that for some vertex $v \in V$, $d^+_{\mathrm{OPT}}(v) \leq \ell_v - 1$. For the rest of the proof, we ignore all vertices that are not in the ℓ_v-core. Now, clearly in G_{ℓ_v} as well we have that $d^+_{\mathrm{OPT}}(v) \leq \ell_v - 1$. On the other hand, we have that every vertex in the unoriented G_{ℓ_v} has degree at least ℓ_v.

Define S to be the set of all vertices that can reach v in G_{ℓ_v} by some directed path of oriented edges. Note that by the simple claim above, for any vertex $u \in S$ we have $d^+_{\mathrm{OPT}}(u) \leq \ell_v - 1 + 1 = \ell_v$. while $d^+_{\mathrm{OPT}}(u)$ itself is strictly less than $\ell_v/2$, so the average load in S is strictly less than $\ell_v/2$.

We will now prove a contradictory claim: S must have some edges directed into S from outside of S. This is contradictory because we defined S to be maximal. To yield the contradiction, let E_S be all edges incident in an undirected sense to S, and let E^*_S be all edges that are oriented outwards from vertices in S, possibly to some other vertex in S. We want to show that $E_S > E^*_S$, which implies that some edge is incoming into S. We know that E^*_S is the sum of the out-degrees of vertices in S, so given the upper bound on the average load in S, we have that $E^*_S < |S| \cdot \ell_v/2$. On the other hand, $E_S \geq |S| \cdot \ell_v/2$, because every vertex in S has degree at least ℓ_v. ◀

References

1 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.

2 Amihood Amir, Tsvi Kopelowitz, Avivit Levy, Seth Pettie, Ely Porat, and B. Riva Shalom. Mind the gap: Essentially optimal algorithms for online dictionary matching with one gap. In *Accepted to International Symposium on Algorithms and Computation (ISAAC)*, 2016.

3 Yossi Azar, Joseph Naor, and Raphael Rom. The competitiveness of on-line assignments. In *SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)*, 1992.

4 Glencora Borradaile, Jennifer Iglesias, Theresa Migler, Antonio Ochoa, Gordon T. Wilfong, and Lisa Zhang. Egalitarian graph orientations. *CoRR*, abs/1212.2178, 2012.

5 Gerth Stølting Brodal and Rolf Fagerberg. Dynamic representation of sparse graphs. In *Algorithms and Data Structures, 6th International Workshop, WADS*, pages 342–351, 1999. doi:10.1007/3-540-48447-7_34.

6 Julie Anne Cain, Peter Sanders, and Nick Wormald. The random graph threshold for k-orientiability and a fast algorithm for optimal multiple-choice allocation. In *18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 469–476. SIAM, 2007. URL: http://dl.acm.org/citation.cfm?id=1283383.1283433.

7 Norishige Chiba and Takao Nishizeki. Arboricity and subgraph listing algorithms. *SIAM J. Comput.*, 14(1):210–223, 1985.

8 Marek Chrobak and David Eppstein. Planar orientations with low out-degree and compaction of adjacency matrices. *Theor. Comput. Sci.*, 86(2):243–266, 1991. doi:10.1016/0304-3975(91)90020-3.

9 Zdenek Dvorak and Vojtech Tuma. A dynamic data structure for counting subgraphs in sparse graphs. In *Algorithms and Data Structures - 13th International Symposium, WADS*, pages 304–315, 2013. doi:10.1007/978-3-642-40104-6_27.

10 David Eisenstat, Philip N. Klein, and Claire Mathieu. An efficient polynomial-time approximation scheme for steiner forest in planar graphs. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 626–638, 2012. URL: http://dl.acm.org/citation.cfm?id=2095116.2095169.

11 David Eppstein. All maximal independent sets and dynamic dominance for sparse graphs. *ACM Transactions on Algorithms*, 5(4), 2009. doi:10.1145/1597036.1597042.

12 Leah Epstein and Asaf Levin. Robust algorithms for preemptive scheduling. In *ESA*, volume 6942 of *Lecture Notes in Comput. Sci.*, pages 567–578. Springer, Heidelberg, 2011. doi:10.1007/978-3-642-23719-5_48.

13 Rudolf Fleischer and Michaela Wahl. Online scheduling revisited. In *Proceedings of the 8th Annual European Symposium on Algorithms*, ESA '00, pages 202–210, London, UK, UK, 2000. Springer-Verlag. URL: http://dl.acm.org/citation.cfm?id=647910.740462.

14 Ashish Goel, Adam Meyerson, and Serge A. Plotkin. Combining fairness with throughput: online routing with multiple objectives. In *STOC*, pages 670–679, 2000.

15 Albert Gu, Anupam Gupta, and Amit Kumar. The power of deferral: maintaining a constant-competitive steiner tree online. In *STOC*, pages 525–534, 2013.

16 Anupam Gupta, Amit Kumar, and Cliff Stein. Maintaining assignments online: Matching, scheduling, and flows. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 468–479, 2014.

17 Meng He, Ganggui Tang, and Norbert Zeh. Orienting dynamic graphs, with applications to maximal matchings and adjacency queries. In *ISAAC*, pages 128–140, 2014.

18 Makoto Imase and Bernard M. Waxman. Dynamic Steiner tree problem. *SIAM J. Discrete Math.*, 4(3):369–384, 1991.

19 Jon M. Kleinberg, Yuval Rabani, and Éva Tardos. Fairness in routing and load balancing. *J. Comput. Syst. Sci.*, 63(1):2–20, 2001.

20 Tsvi Kopelowitz, Robert Krauthgamer, Ely Porat, and Shay Solomon. Orienting fully dynamic graphs with worst-case time bounds. In *ICALP (2)*, pages 532–543, 2014.

21 Tsvi Kopelowitz, Seth Pettie, and Ely Porat. Dynamic set intersection. In *Proceedings 14th Int'l Symposium on Algorithms and Data Structures (WADS)*, pages 470–481, 2015.

22 Tsvi Kopelowitz, Seth Pettie, and Ely Porat. Higher lower bounds from the 3SUM conjecture. In *SODA*, pages 1272–1287, 2016.

23 Lukasz Kowalik. Adjacency queries in dynamic sparse graphs. *Inf. Process. Lett.*, 102(5):191–195, 2007. doi:10.1016/j.ipl.2006.12.006.

24 Lukasz Kowalik and Maciej Kurowski. Oracles for bounded-length shortest paths in planar graphs. *ACM Transactions on Algorithms*, 2(3):335–363, 2006. doi:10.1145/1159892.1159895.

25 R. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *ACM EC*, 2004.

26 Nicole Megow, Martin Skutella, José Verschae, and Andreas Wiese. The power of recourse for online MST and TSP. In *ICALP (1)*, pages 689–700, 2012.

27 Ofer Neiman and Shay Solomon. Simple deterministic algorithms for fully dynamic maximal matching. In *Proceedings of the 45th ACM Symposium on Theory of Computing, STOC*, pages 745–754, 2013. doi:10.1145/2488608.2488703.

28 S. Phillips and J. Westbrook. On-line load balancing and network flow. *Algorithmica*, 21(3):245–261, 1998. doi:10.1007/PL00009214.

29 John F. Rudin, III and R. Chandrasekaran. Improved bounds for the online scheduling problem. *SIAM J. Comput.*, 32(3):717–735, March 2003. doi:10.1137/S0097539702403438.

30 Peter Sanders, Naveen Sivadasan, and Martin Skutella. Online scheduling with bounded migration. *Math. Oper. Res.*, 34(2):481–498, 2009. doi:10.1287/moor.1090.0381.

31 Stephen. B. Seidman. Network structure and minimum degree. *Social Networks*, 5(3):269–287, 1983.

32 Martin Skutella and José Verschae. A robust PTAS for machine covering and packing. In *ESA (I)*, volume 6346 of *LNCS*, pages 36–47. Springer, Berlin, 2010. doi:10.1007/978-3-642-15775-2_4.

33 Jeffery Westbrook. Load balancing for response time. *J. Algorithms*, 35(1):1–16, 2000. doi:10.1006/jagm.2000.1074.

Approximating Approximate Distance Oracles

Michael Dinitz[*1] **and Zeyu Zhang**[†2]

1 **Department of Computer Science, Johns Hopkins University, Baltimore, USA**
mdinitz@cs.jhu.edu
2 **Department of Computer Science, Johns Hopkins University, Baltimore, USA**
zyzhang92@gmail.com

Abstract

Given a finite metric space (V, d), an approximate distance oracle is a data structure which, when queried on two points $u, v \in V$, returns an approximation to the the actual distance between u and v which is within some bounded stretch factor of the true distance. There has been significant work on the tradeoff between the important parameters of approximate distance oracles (and in particular between the size, stretch, and query time), but in this paper we take a different point of view, that of per-instance optimization. If we are given an particular input metric space and stretch bound, can we find the smallest possible approximate distance oracle for that particular input? Since this question is not even well-defined, we restrict our attention to well-known classes of approximate distance oracles, and study whether we can optimize over those classes.

In particular, we give an $O(\log n)$-approximation to the problem of finding the smallest stretch 3 Thorup-Zwick distance oracle, as well as the problem of finding the smallest Pătraşcu-Roditty distance oracle. We also prove a matching $\Omega(\log n)$ lower bound for both problems, and an $\Omega(n^{\frac{1}{k}-\frac{1}{2^{k-1}}})$ integrality gap for the more general stretch $(2k-1)$ Thorup-Zwick distance oracle. We also consider the problem of approximating the best TZ or PR approximate distance oracle *with outliers*, and show that more advanced techniques (SDP relaxations in particular) allow us to optimize even in the presence of outliers.

1998 ACM Subject Classification E.1 Data Structures

Keywords and phrases Distance Oracles, Approximation Algorithms

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.52

1 Introduction

Given a finite metric space (V, d), an approximate distance oracle is a data structure which can approximately answer distance queries. It is usually a combination of a preprocessing algorithm to compute a data structure, and a query algorithm which returns a distance $d'(u, v)$ whenever queried on a pair of vertices $u, v \in V$. An approximate distance oracle is said to have *stretch* t if $d(u, v) \leq d'(u, v) \leq t \cdot d(u, v)$. Note that there is a trivial stretch 1 distance oracle that uses $\Theta(n^2)$ space: we could just store the entire metric space. So the goal is to reduce the space, i.e., to build a small data structure that also has small stretch and small query time.

The seminal work on approximate distance oracles is due to Thorup and Zwick [19]. They showed that for every integer $k \geq 1$, every finite metric space has an approximate distance oracle with stretch $(2k-1)$ and query time $O(k)$ which uses only $O(kn^{1+\frac{1}{k}})$ space.

* Partially supported by NSF grants 1464239 and 1535887.
† Partially supported by NSF grants 1464239.


© Michael Dinitz and Zeyu Zhang;
licensed under Creative Commons License CC-BY

A significant fraction of more recent results have built off of the ideas developed in [19], and much of this follow-up work has stored the exact same (or very similar) data structure, just with improved query algorithms or slightly different information in the storage (see, e.g., [17, 21, 5, 6]). Most notably, Pătraşcu and Roditty [16] gave a different distance oracle (still using some of the basic ideas from [19]) that has multiplicative stretch of 2 and additive stretch of 1, with size $O(n^{\frac{5}{3}})$. This broke through the stretch 3 barrier from [19]. Later this result was improved to more general multiplicative/additive stretches [1].

In this paper we ask a natural but very different type of question about approximate distance oracles: can we find (or approximate) the *best* approximate distance oracle? If we are given an input metric space and a stretch bound, is it possible to find the *smallest* approximate distance oracle for that particular input? This is an unusual question in two ways. First, most data structures are by design forced to store all of the input data; the question is *how* to store it and what *extra* information should be stored. This is the case in other settings where instance-optimality of data structures has been considered, e.g., static or dynamic optimality of splay trees. Second, it is not clear whether this question is even well-defined: lower bounds on data structures are commonly arrived at through information or communication complexity (see, e.g., [15]) but when we ask for the optimal data structure on one particular instance this approach becomes meaningless.

However, approximate distance oracles are different in ways which allow us to make meaningful progress towards these optimization questions. First, since we are allowed to return only approximate distances (up to some stretch factor), we are allowed to store only part of the input (and indeed this is the entire point of such an oracle). The second problem is a bit more tricky: given an input, how can we optimize over "the space of all approximate distance oracles"? What does this mean, and what does this space look like?

To get around this issue, we make an observation: many modern distance oracles (and in particular Thorup-Zwick, Pătraşcu-Roditty, and almost all of their variants) have a similar structure. The preprocessing algorithm chooses a subset of the original distances to store which has some particular structure, and the query algorithm can return a valid distance estimate efficiently as long as the stored distances satisfy the required structure. Thus we can optimize for *these particular* distance oracles by choosing the *best possible* set of distances to remember subject to the required structure. By characterizing this structure for different types of distance oracles, we can optimize over those types.

For example, the stretch-3 Thorup-Zwick distance oracle uses a subtle but simple method to choose the set of distances to store. It randomly samples a subset of approximately $\sqrt{n}$ vertices, without using any information about the original metric space, and then creates a data structure which is related (in a well-defined, important way) to these vertices. The correctness of the query algorithm does not depend on the choice of the vertices. Thus instead of simply choosing the subset of vertices uniformly at random, we can instead try to optimize the set of chosen vertices with respect to the actual input metric space.

In this paper, we give matching $\Theta(\log n)$ upper and lower bounds for optimizing stretch-3 Thorup-Zwick distance oracles, and matching $\Theta(\log n)$ upper and lower bounds for optimizing the Pătraşcu-Roditty distance oracle. These upper bounds both use a similar LP relaxation, but by giving an $\Omega(n^{\frac{1}{k}-\frac{1}{2^{k-1}}})$ integrality gap for optimizing stretch-$(2k-1)$ Thorup-Zwick distance oracles, we show that this relaxation is not enough to give nontrivial approximations when extended to larger stretch values.

As an extension, we also study the problem of optimizing distance oracles *with outliers*: if we are allowed to not answer queries for some of the vertices (of our choosing), can we have much smaller storage space? We give an $(O(\log n), 1+\varepsilon)$-bicriteria approximation to

both stretch-3 Thorup-Zwick and Pătraşcu-Roditty distance oracles with outliers. We also give a true approximation to stretch-3 Thorup-Zwick distance oracle with outliers when the number of outliers is small.

1.1 Relationship to Spanners

It is worth noting that this paper is motivated by a similar line of research on *graph spanners* (subgraphs which approximately preserve distances). Spanners and distance oracles tend to be related (although there is no known formal connection between them), and the traditional questions asked of spanners (what is the tradeoff between the stretch and the size?) are similar to the traditional questions asked of distance oracles. Recently, there has been significant progress in looking at spanners from an optimization point of view: given an input graph and an allowed stretch bound, can we find the sparsest possible spanner meeting that stretch bound? In the last few years, upper and lower bounds have been developed for these problems in the basic case, the directed case, with a degree objective, with fault-tolerance, etc. See, e.g., [9, 2, 11, 8, 7].

It is natural to ask these kinds of optimization questions for distance oracles as well, but the definitions become much more difficult. For spanners, the space we are optimizing over (all subgraphs) is very clear and well-defined. But for distance oracles, as discussed, it is much harder to define the space of all data structures. Thus in this paper we optimize over restricted classes, where this space is more well-defined. We view our definitions of these restricted optimization questions as one of the major contributions of this work.

2 Definitions and Preliminaries

We begin with some basic definitions, including formal definitions of the problems that we will be working on.

▶ **Definition 1.** An approximate distance oracle with (m, a)-stretch, size s, preprocessing time g, and query time h is a pair of algorithms, *preprocess* and *query*, with the following properties.

- *preprocess* is a randomized preprocessing algorithm $preprocess(V, d, m, a, r)$ which takes as input a metric space (V, d), stretch bound (m, a), and random string r and outputs a data structure S where the expected output size is at most $\mathbb{E}_r[|S|] \leq s(|V|, m, a)$ and the expected preprocessing time is at most $g(|V|, m, a)$.
- *query* takes as input a data structure $S = preprocess(V, d, m, a, r)$ (the output of the preprocess algorithm) with two vertices $u, v \in V$, and outputs a value $d'(u, v) \in \mathbb{R}$ such that $d(u, v) \leq d'(u, v) \leq m \cdot d(u, v) + a$. The running time of *query* is at most $h(|V|, m, a)$.

We will frequently refer to these just as "distance oracles" rather than "approximate distance oracles" when the stretch bound is clear from context.

The query algorithm guarantees here are deterministic: the randomness only affects the size of the data structure. Note that one could easily define distance oracles so that either the correctness (with respect to the stretch bound) or the query running time (or both) hold only in expectation or with high probability, but as discussed in Section 1, essentially all existing distance oracles (and in particular the Thorup-Zwick distance oracle) have deterministic guarantees on the queries.

This naturally leads us to the following question: If we fix a particular distance oracle and metric space, can we find the *best* possible data structure? Here we will focus on the output size, not the preprocessing time (as long as the preprocessing time is polynomial). In

other words, since the query algorithm work on *any* of the possible data structures which the preprocessing algorithm might output, can we actually find the smallest such data structure? This gives the following natural optimization problem.

▶ **Definition 2.** Given an approximate distance oracle $\mathcal{A} = (preprocess, query)$, the $\mathcal{A}$-optimization problem takes as input a metric space (V, d) and a stretch bound (m, a), and the goal is to find a string r which minimizes $|preprocess(V, d, m, a, r)|$.

In this paper we will focus on two distance oracles (Thorup-Zwick [19] and Pătraşcu-Roditty [16]), so we now introduce these oracles.

2.1 Thorup-Zwick Distance Oracle

For every integer $k \geq 1$, Thorup and Zwick [19] provided an approximate distance oracle with $(2k-1, 0)$-stretch, size $O(n^{1+\frac{1}{k}})$, preprocessing time $O(kn^{2+\frac{1}{k}})$, and query time $O(k)$. We call this distance oracle TZ_k.

Their preprocessing algorithm first constructs a chain of subsets $\varnothing = A_k \subseteq A_{k-1} \subseteq \ldots \subseteq A_0 = V$ by repeated sampling. Each set A_i, where $i \in [k-1]$, is obtained by including each element of A_{i-1} independently with probability $n^{-\frac{1}{k}}$.

Let $R_{iu} = \{v \in A_{i-1} \mid d(u,v) < \min_{w \in A_i} d(u,w)\}$ for all $u \in V$ and $i \in [k]$ (where by convention $\min_{w \in \varnothing} d(u,w) = \infty$ for all $u \in V$ to handle the $i = k$ case). The output data structure is obtained by storing (in a 2-level hash table) the distance from each node u to each node in $\bigcup_{i=1}^{k} R_{iu}$.

The data structure also stores a little more information. Each vertex u remembers $k-1$ pivots: $\arg\min_{w \in A_i} d(u,w)$ for all $i \in [k-1]$, and the distance from u to these pivots. However, this is a fixed space cost, and also negligible, so when analyzing the size of the oracle we will ignore the cost of storing the pivots

Clearly the output data structure is determined once $A_1, \ldots, A_{k-1}$ are fixed. The size of the data structure is:

$$cost(A_1, \ldots, A_{k-1}, V, d) = \sum_{u \in V} \sum_{i=1}^{k} |R_{iu}| = \sum_{u \in V} \sum_{i=1}^{k} \left| \{v \in A_{i-1} \mid d(u,v) < \min_{w \in A_i} d(u,w)\} \right|.$$

We will refer to $\sum_{u \in V} |R_{iu}|$ as the cost in level i.

Let us look back on the definition of approximate distance oracle. The random string r is only used to generate A_i's, and the query algorithm will return a correct distance estimate no matter what the sets A_i are, but the size is determined by the sets. Therefore, the TZ_k-optimization problem is to find the subsets $\varnothing = A_k \subseteq A_{k-1} \subseteq \ldots \subseteq A_0 = V$ in order to minimize the total cost.

2.2 Pătraşcu-Roditty Distance Oracle

Pătraşcu and Roditty [16] provided an approximate distance oracle with $(2, 1)$-stretch, size $O(n^{\frac{5}{3}})$, preprocessing time $O(n^2)$, and query time $O(1)$. We call this distance oracle PR. Note that PR works only for metric spaces with integer distances.

Their preprocessing algorithm first construct a set $A \subseteq V$ via a complicated correlated sampling (informally, they sample a large set and a small set, and then define A to be everything in the large set and everything contained in a ball around the small set delimited by the large set). The data structure consists of a 2-level hash table for the distance from each node in A to each node in V, as well as a 2-level hash table storing the distance between each pair $\{u, v\} \subseteq V$ such that $d(u,v) < \min_{w \in A} d(u,w) + \min_{w \in A} d(v,w) - 1$.

As with Thorup-Zwick, the output data structure is completely determined once A is fixed. Let $R = \{\{u, v\} \subseteq V \mid d(u, v) < \min_{w \in A} d(u, w) + \min_{w \in A} d(v, w) - 1\}$. Then the size of the data structure is

$$\begin{aligned} cost(A, V, d) &= n \cdot |A| + |R| \\ &= n \cdot |A| + \left| \left\{ \{u, v\} \subseteq V : d(u, v) < \min_{w \in A} d(u, w) + \min_{w \in A} d(v, w) - 1 \right\} \right|. \end{aligned}$$

As before, the random string r is only used to generate the set A, and any $A \subseteq V$ gives a data structure on which the query algorithm works. Therefore, the PR-optimization problem is to find the subset $A \subseteq V$ in order to minimize the total cost.

2.3 Distance Oracles With Outliers

In some cases, a small set of outlier vertices may make the size of the data structure blow up. Yet in some applications it is acceptable to ignore these outliers. This was the motivation behind a line of work on distance oracles with slack ([3], [4]), in which the data structure could ignore the stretch bound on a small fraction of the distances.

In this paper, we consider the case that we can refuse to answer distance queries for some outlier vertices. In other words, we can essentially remove an outlier set F out of V when computing the distance oracle. This gives us the problem of optimizing distance oracle with outliers, in which we not only need to find the random string to determine the output data structure, we also need to find the set of outliers to minimize the final cost. More formally, we have the following type of problem.

▶ **Definition 3.** Given an approximate distance oracle $\mathcal{A} = (preprocess, query)$, the $\mathcal{A}$-optimization problem with outliers takes as input a metric space (V, d), a stretch bound (m, a), and a bound on the number of outliers $f \in \mathbb{N}$. The goal is to find a string r as well as a set $F \subseteq V$ where $|F| \leq f$, in order to minimize $|preprocess(V \backslash F, d, m, a, r)|$.

We will provide both true approximation results and (α, β)-bicriteria results, in which we slightly violate the bound on the number of outliers. Formally, an (α, β)-approximation algorithm for the $\mathcal{A}$-optimization problem with outliers is an algorithm which on any input $((V, d), (m, a), f)$ returns a solution with cost at most $\alpha \cdot OPT$ that has at most $\beta \cdot f$ outliers (where OPT is the minimum cost of any solution with at most f outliers).

2.4 Our Results and Techniques

With these definitions in hand, we can now formally state our results.

In Section 3 we discuss the problem of optimizing the 3-stretch Thorup-Zwick distance oracle, i.e., the TZ_2-optimization problem. It is straightforward to obtain an $O(\log n)$-approximation by reducing to the non-metric facility location problem.

▶ **Theorem 4.** *There is an $O(\log n)$-approximation algorithm for the TZ_2-optimization problem.*

To prove a matching lower bound, we use a reduction from Label Cover to the TZ_2-optimization problem. We use a proof which is similar to the proof of the hardness of Set Cover in [20] (based on [13]). However, we cannot use a reduction directly from Set Cover since we will need some extra properties of the starting instances, and thus are forced to start from Label Cover. We introduce a new notion of (m, l, δ)-set families and show that these can still be plugged into existing hardness results to get the extra structural properties that we need. This lets us prove the following theorem:

▶ **Theorem 5.** *Unless* $\mathbf{NP} \subseteq \mathbf{DTIME}(n^{O(\log\log n)})$*, the* TZ_2*-optimization problem does not admit a polynomial-time* $o(\log n)$*-approximation.*

For larger stretch values, a natural approach is to realize that a simple LP relaxation suffices to give Theorem 4 in the stretch 3 case, and try to extend this basic LP to larger stretches. In Section 4, we show that this does not work for the more general TZ_k-optimization problem: the integrality gap jumps up to become a polynomial. The instance is very simple: it is just the metric space formed by shortest paths on the n-cycle. It turns out to be straightforward to calculate the optimal fractional LP cost, but proving that the optimal integral solution is large is surprisingly involved.

▶ **Theorem 6.** *The basic LP relaxation for the* TZ_k*-optimization problem has an* $\Omega(n^{\frac{1}{k}-\frac{1}{2^{k-1}}})$ *integrality gap when* $k > 2$*.*

In Section 5 we discuss the problem of optimizing the Pătraşcu-Roditty distance oracle. The basic LP and a simple rounding algorithm gives us an $O(\log n)$-approximation algorithm.

▶ **Theorem 7.** *There is an* $O(\log n)$*-approximation algorithm for* PR*-optimization problem.*

A reduction from set cover problem also gives us a matching lower bound.

▶ **Theorem 8.** *Unless* $\mathbf{P} = \mathbf{NP}$*, the* PR*-optimization problem does not admit a polynomial-time* $o(\log n)$*-approximation.*

In Section 6 we move to the outliers setting. For both TZ_2- and PR-optimization problems, a semidefinite programming relaxation and a simple rounding algorithm gives us an $(O(\frac{\log n}{\varepsilon}), 1+\varepsilon)$-approximation algorithm. Here, using an SDP relaxation seems to be necessary – the corresponding LP relaxation requires violating the number of outliers by a factor of 2 rather than a factor of $1+\varepsilon$. We can also get a true approximation on TZ_2-optimization problem with outliers if the number of outliers is low. These results form the following theorems.

▶ **Theorem 9.** *There is an* $(O(\frac{\log n}{\varepsilon}), 1+\varepsilon)$*-approximation algorithm for the* TZ_2*-optimization problem with outliers.*

▶ **Theorem 10.** *There is an* $O(\log n)$*-approximation algorithm for* TZ_2*-optimization problem with outliers if the number of outliers is at most* $\sqrt{n}$*.*

▶ **Theorem 11.** *There is an* $(O(\frac{\log n}{\varepsilon}), 1+\varepsilon)$*-approximation algorithm for the* PR*-optimization problem with outliers.*

3 TZ_2-Optimization Problem

We first give an $O(\log n)$-approximation for TZ_2-optimization (Theorem 4), and follow this with a matching lower bound.

3.1 Upper Bound

We will prove our upper bound by a reduction to the non-metric facility location problem.

▶ **Definition 12.** In the *non-metric facility location* problem we are given a set F of facilities, a set D of clients, an opening cost function $f : F \to \mathbb{R}^+$, and a connection cost function $c : D \times F \to \mathbb{R}^+$. The goal is to find the set $S \subseteq F$ which minimizes $\sum_{i\in S} f(i) + \sum_{i\in D} \min_{j\in S} c(i,j)$ (i.e. the sum of the opening and connection costs).

Non-metric facility location is a classic problem, and much is known about it, including the following upper bound due to Hochbaum.

▶ **Theorem 13** ([14]). *There is a polynomial time algorithm which gives an $O(\log n)$-approximation to the non-metric facility location problem.*

Hochbaum's algorithm is a greedy algorithm, but it is also straightforward to design an algorithm with similar bounds using an LP relaxation. Since it is not necessary we do not present the relaxation here, but generalizations of the relaxation will prove its importance in the more general TZ_k setting (see Section 4).

We now show that the TZ_2-optimization problem is essentially a special case of non-metric facility location problem. First, simple arithmetic manipulation of the cost function of the TZ_2-optimization problem gives the following:

$$\begin{aligned} cost(A_1, V, d) &= \sum_{u \in V} |R_{1u}| + \sum_{u \in V} |R_{2u}| \\ &= \sum_{u \in V} \left| \{v \in V \mid d(u,v) < \min_{w \in A_1} d(u,w)\} \right| + \sum_{u \in V} |\{v \in A_1 \mid d(u,v) < \infty| \\ &= \sum_{u \in V} \left| \{v \in V \mid d(u,v) < \min_{w \in A_1} d(u,w)\} \right| + n|A_1| \\ &= \sum_{w \in A_1} n + \sum_{u \in V} \min_{w \in A_1} |\{v \in V \mid d(u,v) < d(u,w)\}| \,. \end{aligned}$$

Given an instance (V, d) of the TZ_2-optimization problem, we create an instance of non-metric facility location by setting $F = D = V$, opening costs $f(v) = n$ for all $v \in V$, and connection costs $c(u, w) = |\{v \in V \mid d(u, v) < d(u, w)\}|$ for all $u, w \in V$. Then the cost function of the TZ_2-optimization problem is exactly the same as the cost function of non-metric facility location problem. Therefore TZ_2 is a special case of non-metric facility location, which together with Theorem 13 implies Theorem 4.

3.2 Lower Bound

Proving an $\Omega(\log n)$ hardness of approximation (Theorem 5) turns out to be surprisingly difficult. Details appear in the full version [10]; here we provide an informal overview. Technically we reduce directly to TZ_2-optimization from a version of the Label Cover problem that corresponds to applying parallel repetition [18] to 3SAT-5, which is a standard starting point for hardness reductions. Informally, though, we are "really" reducing from Set Cover: given an instance of Set Cover, we show how to create an instance of TZ_2-optimization where the cost of the optimal solution is the same (up to a constant and a polynomial scaling factor). But in order for our reduction to work, we actually need more than just an arbitrary Set Cover instance: we need a version of Set Cover in which it is hard even to cover *most* of the elements, not just all of them.

So we have to also give a new reduction from Label Cover to Set Cover, showing that even this version of Set Cover is hard. It turns out that Feige's reduction [13], reinterpreted by Vazirani [20], essentially already gives us what we need. We just need to analyze it a bit more carefully. In particular, a key component of this reduction is what Vazirani called (m, l)*-set systems*, which can be thought of as nearly-unbiased sample spaces. We generalize this notion to (m, l, δ)-set systems, given in the following definition.

▶ **Definition 14.** A set B (the universe) and a collection of subsets $C_1, \ldots, C_m$ of B form an *(m, l, δ)-set system* if any collection of l sets in $\{C_1, \ldots, C_m, \overline{C_1}, \ldots, \overline{C_m}\}$ whose union contains at least $(1-\delta)|B|$ elements must include both C_i and $\overline{C_i}$ for some i.

An (m, l)-set system is just a $(m, l, 0)$-set system. While not all (m, l)-set systems are (m, l, δ)-set systems for larger δ, the construction of (m, l)-set systems in [20] actually does generalize directly to larger values of δ. With this tool in hand, we follow through the rest of the reduction and it gives us the type of Set Cover instances which we need. Technically our reduction skips this step by going directly from Label Cover to TZ_2-optimization, but generating these kinds of Set Cover instances is intuitively what the first part of the reduction is doing.

4 TZ_k-Optimization Problem

We now move to the more general TZ_k-optimization problem. While we are not able to give nontrivial upper bounds for this problem, we can at least show that the basic LP relaxation (as discussed in Section 3.1) does not give polylogarithmic bounds in this more general setting.

4.1 The LP

Let $B_u(v) = \{w \in V \mid d(u,w) \leq d(u,v)\}$. For every $v \in V$ and $i \in [k]$, let $x_v^{(i)}$ be a variable which is supposed to be an indicator for whether $v \in A_i$. Similarly, for all $u, v \in V$ and $i \in [k]$, let $y_{uv}^{(i)}$ be a variable which is supposed to be an indicator for whether $v \in R_{iu}$. (Recall that $R_{iu} = \{v \in A_{i-1} \mid d(u,v) < \min_{w \in A_i} d(u,w)\}$) We can easily write an LP relaxation for this problem:

$$\begin{array}{lll} (LP_{TZ_k}): \min & \sum_{i=1}^{k} \sum_{u,v \in V} y_{uv}^{(i)} & \\ s.t. & 0 = x_v^{(k)} \leq x_v^{(k-1)} \leq \ldots \leq x_v^{(1)} \leq x_v^{(0)} = 1 & \forall v \in V \\ & y_{uv}^{(i)} \geq x_v^{(i-1)} - \sum_{w \in B_u(v)} x_w^{(i)} & \forall u, v \in V, i \in [k] \\ & y_{uv}^{(i)} \geq 0 & \forall u, v \in V, i \in [k] \end{array}$$

It can easily be shown that this is a valid relaxation (the proof can be found in the full version [10]). When restricted to the special case of $k = 2$, it is not hard to see that this LP is essentially a special case of the basic LP relaxation for non-metric facility location, which can be used to prove the $O(\log n)$ bound of Theorem 4. But for larger values of k the behavior is different, and does not result in a polylogarithmic integrality gap.

4.2 Integrality Gap

The integrality gap instance is quite simple: the metric (V, d) induced by shortest-path distances in a cycle. Slightly more formally, we let $V = [n]$, and use the cycle distance $d(u,v) = \min\{|u-v|, n + \min\{u,v\} - \max\{u,v\}\}$.

Details can be found in the full version [10]. It turns out to be relatively easy to find a fractional solution to LP_{TZ_k} with cost $O(n^{1+\frac{1}{2^{k-1}}})$ on this instance. The tricky part is lower bounding the optimal solution, i.e., showing that the optimal integral solution has cost at least $\Omega(n^{1+\frac{1}{k}})$. Combining these two results gives us an $\Omega(n^{\frac{1}{k}-\frac{1}{2^{k-1}}})$ integrality gap, proving Theorem 6.

5 PR-Optimization Problem

We now move from Thorup-Zwick distance oracles to Pătraşcu-Roditty distance oracles. We show that from an optimization perspective, they are similar to TZ_2 in that we can find matching bounds: an $O(\log n)$-approximation, and $\Omega(\log n)$-hardness.

5.1 Upper Bound

In this section we prove Theorem 7 by using an LP and randomized rounding to give an $O(\log n)$-approximation to the PR-optimization problem.

Let $B_u(v) = \{w \in V \mid d(u,w) \leq d(u,v)\}$, and $B(u,r) = \{w \in V \mid d(u,w) \leq r\}$. We can see $B_u(v) = B(u, d(u,v))$. Now, let x_v be a variable which is supposed to be an indicator for whether $v \in A$, and let y_{uv} be a variable which is supposed to be an indicator for whether $\{u,v\} \in R$ (recall that $R = \{\{u,v\} \subseteq V \mid d(u,v) < \min_{w\in A} d(u,w) + \min_{w \in A} d(v,w) - 1\}$). We can write the following LP relaxation:

$$\begin{array}{rll} (LP_{PR}): \min & \sum_{v\in V} n \cdot x_v + \sum_{\{u,v\}\subseteq V} y_{uv} & \\ s.t. & y_{uv} \geq 1 - \sum_{w \in B(u,r) \cup B(v,d(u,v)-r)} x_w & \forall u,v \in V, \forall r \in [0, d(u,v)] \\ & x_v \in [0,1] & \forall v \in V \\ & y_{uv} \geq 0 & \forall u,v \in V \end{array}$$

At first blush it may not be obvious that the first type of constraint in this LP really captures the characterization of pairs in R. But it is actually not that hard to see that this is a valid relaxation (a formal proof can be found in the full version [10]). Note that while the number of constraints appears to be exponential (recall that we assume integer weights, but not necessarily unit weights, and hence $d(u,v)$ is not necessarily polynomial in the input size), it is in fact possible to solve this LP in polynomial time. We can do this by noting that for each $u,v \in V$, only at most n different value of r actually yield *different* constraints, so we can simply write the constraints for those values.

Our algorithm is relatively straightforward. We first solve LP_{PR} and get an optimal fractional solution (x^*_v, y^*_{uv}). We then use independent randomized rounding, adding each $v \in V$ to A independently with probability $\min\{4 \ln n \cdot x^*_v, 1\}$.

▶ **Lemma 15.** *If $y^*_{uv} \leq \frac{1}{2}$, then the probability that $\{u,v\} \in R$ is at most $\frac{1}{n}$.*

Proof. If $y^*_{uv} \leq \frac{1}{2}$, then the first constraint implies that $\sum_{w \in B(u,r) \cup B(v,d(u,v)-r)} x^*_w \geq \frac{1}{2}$ for all $r \in [0, d(u,v)]$. Therefore, the probability that $A \cap (B(u,r) \cup B(v, d(u,v) - r)) = \varnothing$ for a specific $r \in [0, d(u,v)]$ is at most

$$\prod_{w \in B(u,r) \cup B(v,d(u,v)-r)} (1 - \min\{4 \ln n \cdot x^*_w, 1\}) \leq e^{-\sum_{w \in B(u,r) \cup B(v,d(u,v)-r)} 4 \ln n \cdot x^*_w} \leq \frac{1}{n^2}$$

A union bound over all the different values of r we used in our LP implies that the probability that there exists an $r \in [0, d(u,v)]$ where $A \cap (B(u,r) \cup B(v, d(u,v) - r)) = \varnothing$ is at most $\frac{1}{n^2} \cdot n = \frac{1}{n}$. We claim that the existence of such an r is implied by $\{u,v\} \in R$, and hence the probability that $\{u,v\} \in R$ is at most $\frac{1}{n}$. To see this, suppose that $\{u,v\} \in R$, i.e. suppose that $d(u,v) < \min_{w\in A} d(u,w) + \min_{w \in A} d(v,w) - 1$. Then if we set $r = \min_{w \in A} d(u,w) - 1$, this implies that $\min_{w \in A} d(v,w) > d(u,v) - r$. But then this would imply that no element of A is in $B(u,r) \cup B(v, d(u,v) - r)$. ◀

Let $OPT_{LP_{PR}}$ denote the optimal cost of LP_{PR}. Then the above lemma implies that the expected cost of the rounding algorithm is at most

$$\begin{aligned}\mathbf{E}[n|A| + |R|] &\leq \sum_{v \in V} n \cdot x_v^* \cdot 4 \ln n + 2 \cdot \sum_{u,v \in V} y_{uv}^* + n^2 \cdot \frac{1}{n} \leq O(\log n) \cdot OPT_{LP_{PR}} + n \\ &\leq O(\log n) \cdot OPT\end{aligned}$$

(where we use the fact that $OPT \geq \Omega(n)$). This completes the proof of Theorem 7.

5.2 $\Omega(\log n)$-hardness

We now show a matching hardness bound for the PR-optimization problem by reducing from the Set Cover problem.

Consider a set cover instance $(\mathcal{U}, \mathcal{S})$ where $|\mathcal{U}| + |\mathcal{S}| = n$. For each $e \in \mathcal{U}$, we create a group of vertices G_e where $|G_e| = 3n$. For each $S \in \mathcal{S}$, we also create a group of vertices G_S where $|G_S| = 3n$.

Now we construct the following metric space: $V = (\bigcup_{e \in \mathcal{U}} G_e) \cup (\bigcup_{S \in \mathcal{S}} G_S)$ and

$$d(u,v) = \begin{cases} 1, & \text{if } u \in G_e, v \in G_e \\ 1, & \text{if } u \in G_S, v \in G_S \\ 1, & \text{if } u \in G_e, v \in G_S, e \in S \\ 2, & \text{otherwise.} \end{cases}$$

In the full version [10] we show that if there is a solution $\mathcal{S}^*$ to the set cover instance $(\mathcal{U}, \mathcal{S})$ where $|\mathcal{S}^*| = t$, then there is a set A where $cost(A, V, d) \leq t|V|$. We also show that if there is a set $A \subseteq V$ where $cost(A, V, d) \leq t|V|$, then there exists a solution $\mathcal{S}^*$ to the set cover instance $(\mathcal{U}, \mathcal{S})$ where $|\mathcal{S}^*| = t$. These two claims, together with an appropriate hardness theorem for Set Cover [12], imply Theorem 8.

6 Distance Oracles With Outliers

We now move to the more difficult outliers setting, where we can also optimize over a set of vertices to ignore. Recall that for an approximate distance oracle $\mathcal{A}$, our goal is now to find a set of vertices F (the outliers) where $|F| \leq f$ as well as a string r in order to minimize $|preprocess(V \setminus F, d, m, a, r)|$. In other words, we are going to try to solve the same problems as before, but where we can choose a set F to remove. We begin with TZ_2, and then move to PR.

6.1 TZ_2-Optimization Problem With Outliers

For this problem, it is easy to see that the cost function becomes:

$$\begin{aligned}cost(A, F, V, d) &= (n - f)|A| + \sum_{u \in V \setminus F} |R_{1u}| \\ &= (n - f)|A| + \sum_{u \in V \setminus F} \left| \{v \in V \setminus F \mid d(u,v) < \min_{w \in A} d(u,w)\} \right|.\end{aligned}$$

A natural approach is to use an LP which is similar to LP_{TZ_k} to solve this problem (but for TZ_2), suitably adapted to handle outliers. Let x_v be a variable which is supposed to be

an indicator for whether $v \in A$, let y_{uv} be a variable which is supposed to be an indicator for whether $v \in R_{1u}$, and let z_v be a variable which is supposed to be an indicator for whether $v \in F$. Then we can write the following natural LP relaxation:

$$\begin{array}{rll} (LP_{TZ_2O}): \min & \sum_{v\in V}(n-f)\cdot x_v + \sum_{u,v\in V} y_{uv} & \\ s.t. & y_{uv} \geq 1 - z_u - z_v - \sum_{w\in B_u(v)} x_w & \forall u,v \in V \\ & \sum_{v\in V} z_v \leq f & \\ & x_v \in [0,1] & \forall v \in V \\ & y_{uv} \geq 0 & \forall u,v \in V \\ & z_v \in [0,1] & \forall v \in V \end{array}$$

Unfortunately, this LP can not give an (α, β)-approximation with $\beta = 2 - \epsilon$. To see this, consider the case that $f = \frac{n}{2}$. Then the optimal solution to LP_{TZ_2O} is 0, by setting all z_v to $\frac{1}{2}$, all x_v to 0, and all $y_{uv} = 0$. Thus any integral solution, to be competitive with this fractional solution, must treat *all* nodes as outliers, requiring β to be at least 2.

Fortunately we can give a stronger semidefinite programming relaxation, allowing for a better approximation. As in LP_{TZ_2O}, let $\vec{x}_v$ be a variable which is supposed to be an indicator for whether $v \in A$, let $\vec{y}_{uv}$ be a variable which is supposed to be an indicator for whether $v \in R_{1u}$, and let $\vec{z}_v$ be a variable which is supposed to be an indicator for whether $v \in F$. We can then write this SDP:

$$\begin{array}{rll} (SDP_{TZ_2O}): \min & \sum_{v\in V}(n-f)\cdot \|\vec{x}_v\|^2 + \sum_{u,v\in V} \|\vec{y}_{uv}\|^2 & \\ s.t. & \|\vec{y}_{uv}\|^2 \geq 1 - \vec{z}_u \cdot \vec{z}_v - \sum_{w\in B_u(v)} \|\vec{x}_w\|^2 & \forall u,v \in V \\ & \sum_{v\in V} \|\vec{z}_v\|^2 \leq f & \\ & \|\vec{x}_v\|^2 \leq 1 & \forall v \in V \\ & \|\vec{y}_{uv}\|^2 \leq 1 & \forall u,v \in V \\ & \|\vec{z}_v\|^2 \leq 1 & \forall v \in V \end{array}$$

Our approximation algorithm first solves SDP_{TZ_2O} to get an optimal solution $(\vec{x}^*_v, \vec{y}^*_{uv}, \vec{z}^*_v)$. We then use independent randomized rounding to construct A, adding each $v \in V$ to A independently with probability $\min\{\frac{3\ln n}{\varepsilon} \cdot \|\vec{x}^*_v\|^2, 1\}$ where ε is a small constant. Finally, we use threshold rounding to construct F by adding each $v \in V$ to F if $\|\vec{z}^*_v\|^2 \geq \frac{1}{1+\varepsilon}$.

We want to show that this is an $(O(\log n), 1+\varepsilon)$-approximation. It is easy to see that $|F| \leq (1+\varepsilon)f$ because $\sum_{v\in V} \|\vec{z}^*_v\|^2 \leq f$. In order to prove Theorem 9 it only remains to prove that the expected cost is at most $O(\log n) \cdot OPT$. This proof can be found in the full version [10].

When $f \leq \sqrt{n}$ we can actually give a true $O(\log n)$-approximation (Theorem 10). The algorithm is almost the same; we just need to change the threshold rounding for outliers to instead pick the f vertices with largest $\|\vec{z}_v\|^2$ value. Details appear in the full version [10].

6.2 PR-Optimization Problem With Outliers

For this problem, the cost function becomes:

$$\begin{aligned} cost(A,F,V,d) &= (n-f)\cdot|A| + |R| \\ &= (n-f)\cdot|A| + \left|\{\{u,v\} \subseteq V\backslash F \mid d(u,v) < \min_{w\in A} d(u,w) + \min_{w\in A} d(v,w) - 1\}\right|. \end{aligned}$$

We will again use an SDP relaxation. Let $\vec{x}_v$ be a variable which is supposed to be an indicator for whether $v \in A$, let $\vec{y}_{uv}$ be a variable which is supposed to be an indicator for whether $\{u,v\} \in R$, and let $\vec{z}_v$ be a variable which is supposed to be an indicator for whether

$v \in F$. We have the following relaxation (which we call SDP_{PR}) which is similar to both LP_{PR} and SDP_{TZ_2O}:

$$\begin{array}{lll} \min & \sum_{v\in V}(n-f)\cdot\|\vec{x}_v\|^2 + \sum_{\{u,v\}\subseteq V}\|\vec{y}_{uv}\|^2 & \\ s.t. & \|\vec{y}_{uv}\|^2 \geq 1 - \vec{z}_u\cdot\vec{z}_v - \sum_{w\in B(u,r)\cup B(v,d(u,v)-r)}\|\vec{x}_w\|^2 & \forall u,v\in V, r\in[0,d(u,v)] \\ & \sum_{v\in V}\|\vec{z}_v\|^2 \leq f & \\ & \|\vec{x}_v\|^2 \leq 1 & \forall v\in V \\ & \|\vec{y}_{uv}\|^2 \leq 1 & \forall u,v\in V \\ & \|\vec{z}_v\|^2 \leq 1 & \forall v\in V \end{array}$$

Note that SDP_{PR} is solvable in polynimial time for the same reason that LP_{PR} is solvable: for each pair of (u,v), we can find n different values of r that give all of the distinct constraints.

The rounding algorithm is basically the same as the TZ_2-optimization problem with outliers. We first solve the SDP_{PR} and get an optimal solution $(\vec{x}^*_v, \vec{y}^*_{uv}, \vec{z}^*_v)$. We then use independent randomized rounding to get A, adding each $v \in V$ to A independently with probability $\min\{\frac{6\ln n}{\varepsilon}\cdot\|\vec{x}^*_v\|^2, 1\}$ where ε is a small constant. Then we use threshold rounding to get F, adding each $v \in V$ to F if $\|\vec{z}^*_v\|^2 \geq \frac{1}{1+\varepsilon}$.

This is an $(O(\log n), 1+\varepsilon)$-approximation. It is easy to see that $|F| \leq (1+\varepsilon)f$ because $\sum_{v\in V}\|\vec{z}^*_v\|^2 \leq f$. The proof that the expected cost is at most $O(\log n)\cdot OPT$ is in the full version [10], which completes the proof of Theorem 11.

7 Conclusion and Future Work

In this paper we initiate the study of *approximating* approximate distance oracles. This is a different take on the question of optimizing data structures, where we attempt to find the best data structure for a particular input, rather than for a class of inputs. In order to make this tractable (or even well-defined), we restrict our attention to known classes of distance oracles, and show that it is sometimes possible to find the best of these restricted oracles. We also extended our approaches to optimize in the presence of outliers.

For future work, the major question is clearly whether we can approximately optimize higher level (i.e., higher stretch) Thorup-Zwick distance oracles. Although we show an integrality gap for the basic LP, it is quite conceivable that a stronger LP or SDP could be used to give a logarithmic approximation ratio. Beyond this, there are other distance oracles which could be optimized – we chose Thorup-Zwick and Pătraşcu-Roditty since they are well-known and in some ways canonical, but it would be interesting to extend these ideas to other oracles. At a higher level, we believe that the definitions and ideas we have introduced here could lead to many interesting questions about optimizing data structures for given inputs: can we find near-optimal distance labels? Or compact routing schemes? Or connectivity oracles? Or fault-tolerant oracles? Essentially any data structure question in which there is a choice of *which* data to store, rather than how to store it, can be put into our optimization framework. Exploring this space is an exciting future direction.

References

1 Ittai Abraham and Cyril Gavoille. On approximate distance labels and routing schemes with affine stretch. In *Proceedings of the International Symposium on Distributed Computing (DISC)*, pages 404–415. Springer, 2011.

2 Piotr Berman, Arnab Bhattacharyya, Konstantin Makarychev, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Approximation algorithms for spanner problems and directed steiner forest. *Information and Computation*, 222:93–107, 2013.

3 T-H Hubert Chan, Kedar Dhamdhere, Anupam Gupta, Jon Kleinberg, and Aleksandrs Slivkins. Metric embeddings with relaxed guarantees. *SIAM Journal on Computing*, 38(6):2303–2329, 2009.

4 T-H Hubert Chan, Michael Dinitz, and Anupam Gupta. Spanners with slack. In *European Symposium on Algorithms*, pages 196–207. Springer, 2006.

5 Shiri Chechik. Approximate distance oracles with constant query time. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 654–663. ACM, 2014.

6 Shiri Chechik. Approximate distance oracles with improved bounds. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 1–10. ACM, 2015.

7 Eden Chlamtác and Michael Dinitz. Lowest degree k-spanner: Approximation and hardness. In *Proceedings of the 17th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, volume 28, pages 80–95. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014.

8 Eden Chlamtac, Michael Dinitz, and Robert Krauthgamer. Everywhere-sparse spanners via dense subgraphs. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 758–767. IEEE Computer Society, 2012. doi:10.1109/FOCS.2012.61.

9 Michael Dinitz and Robert Krauthgamer. Directed spanners via flow-based linear programs. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, pages 323–332. ACM, 2011.

10 Michael Dinitz and Zeyu Zhang. Approximating approximate distance oracles. Full version. URL: https://arxiv.org/abs/1612.05623.

11 Michael Dinitz and Zeyu Zhang. Approximating low-stretch spanners. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 821–840. SIAM, 2016.

12 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 624–633. ACM, 2014.

13 Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

14 Dorit S Hochbaum. Heuristics for the fixed cost median problem. *Mathematical programming*, 22(1):148–162, 1982.

15 Guy Joseph Jacobson. *Succinct Static Data Structures*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 1988.

16 Mihai Patrascu and Liam Roditty. Distance oracles beyond the thorup-zwick bound. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 815–823. IEEE, 2010.

17 Mihai Patrascu, Liam Roditty, and Mikkel Thorup. A new infinity of distance oracles for sparse graphs. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 738–747. IEEE, 2012.

18 Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

19 Mikkel Thorup and Uri Zwick. Approximate distance oracles. *Journal of the ACM (JACM)*, 52(1):1–24, 2005.

20 Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.

21 Christian Wulff-Nilsen. Approximate distance oracles with improved query time. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 539–549. Society for Industrial and Applied Mathematics, 2013.

Fast Cross-Polytope Locality-Sensitive Hashing

Christopher Kennedy[*1] **and Rachel Ward**[†2]

1 **Department of Mathetmatics, University of Texas at Austin, USA**
ckennedy@math.utexas.edu
2 **Department of Mathetmatics, University of Texas at Austin, USA**
rward@math.utexas.edu

Abstract

We provide a variant of cross-polytope locality sensitive hashing with respect to angular distance which is provably optimal in asymptotic sensitivity and enjoys $\mathcal{O}(d \ln d)$ hash computation time. Building on a recent result in [4], we show that optimal asymptotic sensitivity for cross-polytope LSH is retained even when the dense Gaussian matrix is replaced by a fast Johnson-Lindenstrauss transform followed by discrete pseudo-rotation, reducing the hash computation time from $\mathcal{O}(d^2)$ to $\mathcal{O}(d \ln d)$. Moreover, our scheme achieves the optimal *rate of convergence* for sensitivity. By incorporating a low-randomness Johnson-Lindenstrauss transform, our scheme can be modified to require only $\mathcal{O}(\ln^9(d))$ random bits.

1998 ACM Subject Classification F.2.2 Geometrical problems and computations

Keywords and phrases Locality-sensitive hashing, Dimension reduction, Johnson-Lindenstrauss tranform

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.53

1 Introduction

The nearest neighbor search problem is an essential algorithmic component to a wide variety of applications including data compression, information retrieval, image storage, computer vision, and pattern recognition. **Nearest neighbor search (NN)** can be stated as follows: given a metric space $(X, \mathcal{D})$ and a set of points $P = \{x_1, ..., x_n\} \subset X$, for a query point $x \in P$ find $y = \text{argmin}_{x_i \in P \setminus \{x\}} \mathcal{D}(x_i, x)$. In high dimensions, it is known that existing algorithms have poor performance (see [23]); that is, for a query point $x \in P$, any algorithm for NN must essentially compute the distances between x and each point in $P \setminus \{x\}$.

In order to improve on linear search, one may relax the problem to that of *approximate* nearest neighbors search. Precisely, the (R, c) **near neighbor problem (**(R, c)**-NN)** as introduced in [12] is as follows: given a query point $x \in P$ and the assurance of a point $y' \in P$ such that $\mathcal{D}(y', x) < R$, find $y \in P$ such that $\mathcal{D}(y, x) < cR$. In contrast to exact nearest neighbors search, the approximate nearest neighbor search problem can be solved in *sublinear* query time, and this is achieved using **locality sensitive hashing** (LSH). The idea in LSH is to specify a function from the domain X to a discrete set of hash values – a so-called *hash function* – which sends closer points to the same *hash value* with higher probability than points which are far apart. Then, for a set of points $P = \{x_1, ..., x_n\} \subset X$ and a query point $x \in P$, search within its corresponding hash bucket for a nearest neighbor.

* C. Kennedy was supported in part by R. Ward's NSF CAREER grant and an ASOFR Young Investigator Award.
† R. Ward was supported in part by an NSF CAREER grant and an ASOFR Young Investigator Award.


© Christopher G. Kennedy and Rachel A. Ward;
licensed under Creative Commons License CC-BY

From here on out, we fix the space $X = S^{d-1}$ endowed with the euclidean metric. We begin by recalling the standard notion of sensitivity for a hash family; intuitively, a hash family with higher sensitivity is much more likely to hash points that are close to the same hash value, and thus be a better candidate for locality sensitive hashing.

▶ **Definition 1.** For $r_1 \leq r_2$ and $p_2 \leq p_1$, a hash family $\mathcal{H}$ is (r_1, r_2, p_1, p_2)**-sensitive** if for all $x, y \in S^{d-1}$,

- If $\|x - y\|_2 \leq r_1$, then $\Pr_{\mathcal{H}}[h(x) = h(y)] \geq p_1$.
- If $\|x - y\|_2 \geq r_2$, then $\Pr_{\mathcal{H}}[h(x) = h(y)] \leq p_2$.

We primarily care about the case where $r_1 = R$, $r_2 = cR$, and to quantify sensitivity of a certain scheme, we study the parameter

$$\rho = \frac{\ln(p_1^{-1})}{\ln(p_2^{-1})}. \tag{1}$$

The key result linking the sensitivity of a hash family to its performance for $(R, c) - NN$ search is the following:[1]

▶ **Proposition 2** (Theorem 5 in [12])**.** *Given an (R, cR, p_1, p_2)-sensitive hash family $\mathcal{H}$, there exists a data structure that solves $(R, c) - NN$ with constant probability using $\mathcal{O}(dn + n^{1+\rho})$ space, $\mathcal{O}(n^\rho)$ query time, and $\mathcal{O}(n^\rho \ln_{1/p_1} n)$ evaluations of hash functions from $\mathcal{H}$.*

Since the parameter ρ quantifies the performance of a given LSH algorithm for $(R, c)-NN$, it is of interest to make this parameter as small as possible. It was shown in [20] that $\rho = \frac{1}{c^2}$ is asymptotically (in d) optimal for the case of unit sphere with the euclidean metric. Spherical LSH ([5], [6]) was shown to achieve this optimal sensitivity; however, the corresponding hash functions in spherical LSH are not practical to compute. Subsequently, Andoni, Indyk, Laarhoven, and Razenshteyn [4] showed the existence of an LSH scheme with optimally sensitive hash functions which are practical to implement; namely, the *cross-polytope* LSH scheme which has been previously proposed in [22] (see also [7], [20], [19]). Given a matrix $\mathcal{G} \in \mathbb{R}^{d \times d}$ with i.i.d. $\mathcal{N}(0, 1)$ entries, the cross polytope hash of a point $x \in S^{d-1}$ is defined as

$$h(x) = \operatorname*{argmin}_{u = \{\pm e_i\}} \left\| \frac{\mathcal{G}x}{\|\mathcal{G}x\|_2} - u \right\|_2, \tag{2}$$

where $\{e_i\}_{i=1}^d$ is the standard basis for $\mathbb{R}^d$. The paper [4] provided the following collision probability for cross-polytope LSH.

▶ **Proposition 3** (Theorem 1 in [4])**.** *Suppose $x, y \in S^{d-1}$ are such that $\|x - y\|_2 = R$, with $0 < R < 2$, and $\mathcal{H}$ is the hash family defined in (2). Then,*

$$\ln\left(\frac{1}{\Pr_{\mathcal{H}}[h(x) = h(y)]}\right) = \frac{R^2}{4 - R^2} \ln d + \mathcal{O}_R(\ln(\ln d)). \tag{3}$$

Consequently,

$$\rho = \frac{1}{c^2} \frac{4 - c^2 R^2}{4 - R^2} + o(1),$$

[1] In particular, the algorithm stores L hash tables from the family $\mathcal{G}$, where each $g \sim \mathcal{G}$ is given by $g(x) = (h_1(x), ..., h_k(x))$, and $h_i \sim \mathcal{H}, i = 1...k$. Then, given a query point $x \in S^{d-1}$, the algorithm looks for collisions in the buckets $g_1(x), ..., g_L(x)$. The choice of parameters $k = n^\rho$, $L = \ln_{1/p_1} n$ ensure that the algorithm solves $(R, c) - NN$ with constant probability.

where here and in the sequel, $o(1)$ means a parameter that goes to 0 as $d \to \infty$. This implies that the above scheme is asymptotically optimal with respect to ρ.[2] Still, this scheme is limited in efficiency by the $\mathcal{O}(d^2)$ computation required to compute a dense matrix-vector multiplication in (2). To reduce this computation, [4] proposed to to use a pseudo-random rotation in place of a dense Gaussian matrix, namely,

$$h(x) = \text{argmin}_{u=\{\pm e_i\}} \|HD_bHD_{b'}HD_{b''}x - u\|_2 , \tag{4}$$

where $H \in \mathbb{R}^{d\times d}$ is a Hadamard matrix and $D_b, D_{b'}, D_{b''} \in \mathbb{R}^{d\times d}$ are independent diagonal matrices with i.i.d. Rademacher entries on the diagonal. This scheme has the advantage of computing hash functions in time $\mathcal{O}(d \ln d)$, and was shown in [4] to *empirically* exhibit similar collision probabilities to cross-polytope LSH, but provable guarantees on the asymptotic sensitivity of this fast variant of the standard cross-polytope LSH remain open.

1.1 Our Contributions

1.1.1 Fast cross-polytope LSH with optimal asymptotic sensitivity

While we do not prove theoretical guarantees regarding the asymptotic sensitivity of the particular fast variant (4), we construct a different variant of the standard cross-polytope LSH (defined below in (5)) which also enjoys $\mathcal{O}(d \ln d)$ matrix-vector multiplication, and for which we are able to prove optimal asymptotic sensitivity $\rho = \frac{1}{c^2}$:

$$h_F(x) = \underset{u=\{\pm e_i\}}{\text{argmin}} \left\| \frac{\mathcal{G}(H_S D_b x)}{\|\mathcal{G}(H_S D_b x)\|_2} - u \right\|_2 ; \tag{5}$$

Here, $D_b : \mathbb{R}^d \to \mathbb{R}^d$ is a diagonal matrix with i.i.d. Rademacher entries on the diagonal, $H_S \in \mathbb{R}^{m\times d}$ is a partial Hadamard matrix restricted to a random subset $S \subset [d]$ of $|S| = m = \mathcal{O}(\log(d))$ rows, and $\mathcal{G} : \mathbb{R}^m \to \mathbb{R}^{d'}$ is a Gaussian matrix that lifts and rotates in dimension d' in the range $m \le d' \le d$. There is nothing special about lifting to dimension d, and indeed one could lift to dimension $d' > d$, but if d' grows faster than d, the hash computation no longer takes time $\mathcal{O}(d \ln d)$.

The embedding $H_S D_b x$ acts as a Johnson-Lindenstrauss (JL) transform[3], and embeds the points in dimension $m \approx \ln d$.

It is straightforward that the hash computation $x \to h_F(x)$ takes $\mathcal{O}(d'm)$ time from the Gaussian matrix multiplication and $\mathcal{O}(d \ln d)$ time from the JL transform. We will show that optimal asymptotic sensitivity is still achieved without lifting, $d' = m$, but we observe both empirically and theoretically that the *rate of convergence* to the asymptotic sensitivity improves by lifting to higher dimension; taking d' closer to d results in empirically closer results to the standard cross-polytope scheme (see section 5 for more details). Moreover, our scheme achieves the lower bound given by Theorem 2 in [4] for the fastest rate of convergence among all hash families which has to d' values.

---

[2] In fact, the coefficient $\frac{4-c^2R^2}{4-R^2} < 1$ for every choice of $c > 1$ and $0 < R < 2$, but this does not break the lower bound given in [20] since the lower bound $\rho = \frac{1}{c^2}$ only holds for a particular sequence $R = R(d)$. For cross-polytope LSH and the schemes proposed here, any sequence $R(d) \to 0$ suffices.

[3] Formally, given a finite metric space $(X, \|\cdot\|) \subset \mathbb{R}^d$, a JL transform is a linear map $\Phi : \mathbb{R}^d \to \mathbb{R}^m$ such that for all $x \in X$, $(1-\delta)\|x\|^2 \le \|\Phi x\|^2 \le (1+\delta)\|x\|^2$, with $m \ll d$ close to the optimal scaling $m = C\delta^{-2} \ln(|X|)$ [13, 2, 16].

Table 1 Various LSH Families and corresponding Hash Functions.

LSH Family	Hash Function
Cross-Polytope LSH	$h(x) = \underset{u=\{\pm e_i\}}{\operatorname{argmin}} \left\| \frac{\mathcal{G}x}{\|\mathcal{G}x\|_2} - u \right\|_2, \quad \mathcal{G} \in \mathbb{R}^{d\times d}$
Fast Cross-Polytope LSH	$h_F(x) = \underset{u=\{\pm e_i\}}{\operatorname{argmin}} \left\| \frac{\mathcal{G}(H_S D_b x)}{\|\mathcal{G}(H_s D_b x)\|_2} - u \right\|_2, \quad \mathcal{G} \in \mathbb{R}^{d'\times m}$
Fast Discrete Cross-Polytope LSH	$h_D(x) = \underset{u=\{\pm e_i\}}{\operatorname{argmin}} \left\| \frac{\widehat{\mathcal{G}}(Zx)}{\|\widehat{\mathcal{G}}(Zx)\|_2} - u \right\|_2, \quad \widehat{\mathcal{G}} \in \mathbb{R}^{d'\times m}$

1.1.2 Fast cross-polytope LSH with optimal asymptotic sensitivity and few random bits

Aiming to construct a hash family with similar guarantees which also uses as little randomness as possible, we also consider a discretized version of the fast hashing scheme (5) in which the Gaussian matrix $\mathcal{G} \in \mathbb{R}^{d'\times m}$ is replaced by a matrix $\widehat{\mathcal{G}} \in \mathbb{R}^{d'\times m}$ whose entries are i.i.d. discrete approximations of a Gaussian; in place of the "standard" fast JL transform $H_S D_b$, we consider $Z \in \mathbb{R}^{d\times m}$ a low-randomness JL transform that we will clarify later. Then, the discrete fast hashing scheme we consider is

$$h_D(x) = \underset{u=\{\pm e_i\}}{\operatorname{argmin}} \left\| \frac{\widehat{\mathcal{G}}(Zx)}{\|\widehat{\mathcal{G}}(Zx)\|_2} - u \right\|_2. \tag{6}$$

Also for this scheme, the hash computation $x \to h(x)$ takes $\mathcal{O}(d'm)$ time from the Gaussian matrix multiplication and $\mathcal{O}(d \ln d)$ time from the JL transform. Our scheme has several advantages, due to the fact that the choice of d' in the range $d \le d' \le m$ is flexible: To summarize our main contributions, we prove for both the fast cross-polytope LSH and the fast discrete cross-polytope LSH,

- For each d' in the range $m \le d' \le d$, this scheme achieves the asymptotically optimal ρ. Moreover, for $d' = d$, the rate of convergence to this ρ is optimal over all hash families with d hash values.
- With the choice $d' = d$, the scheme computes hashes in time $\mathcal{O}(d \ln d)$ and performs well empirically compared to the standard cross-polytope with dense Gaussian matrix (see section 5).
- With the choice $d' = m$, and by discretizing the Gaussian matrix, we arrive at a scheme that has only $\mathcal{O}(\ln^9(d))$ bits of randomness and still has optimal asymptotic sensitivity.

Table 1 contains the construction of the original cross-polytope LSH scheme, our fast cross-polytope scheme, as well as the discretized version.

1.2 Related work

Many of our results hinge on the careful analysis of collision probabilities for the cross-polytope LSH scheme given in [4]. Additionally, various ways to reduce the runtime of cross-polytope LSH, specifically using fast, structured projection matrices, are mentioned

in [8]. They also define a generalization of cross-polytope lsh that first projects to a low dimensional subspace, but they never consider lifting back up to a high dimensional subspace again. Johnson-Lindenstrauss transforms have previously been used in many approximate nearest neighbors algorithms, (see [12], [18], [1], [21], [5], and [9], to name a few), primarily as a preprocessing step to speed up computations that have some dependence on the dimension. LSH with p-stable distributions, as introduced in [10], uses a random projection onto a single dimension, which is later generalized in [3] to random projection onto $o(\ln d)$ dimensions, with the latter having optimal exponent $\rho = \frac{1}{c^2} + \mathcal{O}(\ln(\ln d)/\ln^{1/3} d)$. We make a note that our scheme uses dimension reduction slightly differently, as an intermediate step before lifting the vectors back up to a different dimension.

Similar dimension reduction techniques have been used in [17], where the data is sparsified and then a random projection matrix is applied. The authors exploit the fact that the random projection matrix will have the restricted isometry property, which preserves pairwise distances between any two sparse vectors. This result is notable in that the reduced dimension has no dependence on n, the number of points. See section 4 for more discussion.

2 Notation

We now establish notation that will be used in the remainder. $\mathcal{O}_R(f(d))$ is to mean $\mathcal{O}_R(f(d)) = \mathcal{O}(f(d)g(R))$ for some finite valued function $g : (0, 2) \to \mathbb{R}$. The expression $o(1)$ is a quantity such that $\lim_{d\to\infty} o(1) = 0$. $H \in \mathbb{R}^{d\times d}$ is the Hadamard matrix. $D_b \in \mathbb{R}^{d\times d}$ is a diagonal matrix whose entries are i.i.d. Rademacher variables. For a matrix $M \in \mathbb{R}^{d\times d}$, M_S will denote the restriction of M to its rows indexed by the set $S \subset \{1, ..., d\}$. The variable $\mathcal{G}$ will always denote a matrix with i.i.d. standard normal Gaussian entries, where the matrix may vary in size. The variable $\widehat{\mathcal{G}}$ will always denote a matrix with i.i.d. copies of a discrete random variable X which roughly models a Gaussian. C will denote various constants that are bounded independent of the dimension. We will use m to denote the projected dimension of our points, where $m \ll d$, and d' the lifted dimension, where $m \leq d' \leq d$. For a vector $x \in S^{d-1}$ we will denote $\tilde{x} = H_S D_b x$.

3 Main Results

We now formalize the intuition about how our scheme behaves relative to cross-polytope LSH.

▶ **Theorem 4.** *Suppose $\mathcal{H}$ is the family of hash functions defined in (5) with the choice $m = \mathcal{O}(\ln^5(d)\ln^4(\ln d))$, and ρ is as defined in (1) for this particular family. Then we have (i-)*

$$\rho = \frac{1}{c^2}\frac{4 - c^2R^2}{4 - R^2} + o(1).$$

and this hashing scheme runs in time $\mathcal{O}(d \ln d)$.

Moreover, we have the optimal rate of convergence,
(ii-)

$$\rho = \frac{1}{c^2}\frac{4 - c^2R^2}{4 - R^2} + \mathcal{O}\left(\frac{1}{\ln d'}\right).$$

The lower bound given by Theorem 2 in [4] verifies the above rate of convergence is in fact optimal. We remark that when hashing n points simultaneously, the embedded dimension m

picks up a factor of $\ln(n)$. Assuming that n is polynomial in d, the result in Theorem 4 still holds simultaneously over all pairs of points.

In addition to creating a fast hashing scheme, one can reduce the amount of randomness involved. In particular, we show that a slight alteration of the scheme still achieves the optimal ρ-value while using only $\mathcal{O}(\ln^9 d)$ bits of randomness. The idea is to replace the Gaussian matrix by a matrix of i.i.d. discrete random variables. Some care is required in tuning the size of this matrix so that the correct number of bits is achieved. As a consequence the number of hash values for this scheme is of order $\mathcal{O}(m)$ (i.e. we lift up to a smaller dimension), which lowers performance in practice, but does not affect the asymptotic sensitivity ρ. We additionally use a JL transform developed by Kane and Nelson [14] that only uses $\mathcal{O}(\ln(d)\ln(\ln d))$ bits of randomness. Specifically, the hash function for this scheme is

$$h_D(x) = \operatorname*{argmin}_{u=\{\pm e_i\}} \left\| \frac{\widehat{\mathcal{G}}(Zx)}{\|\widehat{\mathcal{G}}(Zx)\|_2} - u \right\|_2$$

where $\widehat{\mathcal{G}} \in \mathbb{R}^{d' \times m}$ is a matrix with i.i.d. copies of a discrete random variable X which roughly models a Gaussian, and $Z \in \mathbb{R}^{d \times m}$ is the JL transform constructed in [14]. Our analysis allows us to pick the threshold value $d' = m$ to minimize the number of random bits.

▶ **Theorem 5.** *There is a hash family $\mathcal{H}$ with $\mathcal{O}(\ln^9 d)$ bits of randomness that achieves the bound*

$$\rho = \frac{1}{c^2}\frac{4 - c^2R^2}{4 - R^2} + o(1),$$

and runs in time $\mathcal{O}(d \ln d)$.

3.1 Theorem 4 Part (i-) Proof Outline

First we state an elementary limit result that we will apply to the proofs of both Theorem 4 and Theorem 5.

▶ **Lemma 6.** *Suppose $m_d(a), m_d(b)$ are positive functions, $\lim_{d\to\infty} m_d(a) = a$, $\lim_{d\to\infty} m_d(b) = b$, and that $f(d), g(d)$ are also positive, $\lim_{d\to\infty} f(d) = \lim_{d\to\infty} g(d) = \infty$, $\lim_{d\to\infty} \frac{f(d)}{g(d)} = \infty$. Then,*

$$\lim_{d\to\infty} \frac{m_d(a)f(d) + g(d)}{m_d(b)f(d) + g(d)} = \frac{a}{b}$$

Proceeding to the proof of Theorem 4, the key observation is that for $x, y \in S^{d-1}$, $\mathcal{G}\tilde{x} = \mathcal{G}_0 \begin{bmatrix} \tilde{x} \\ 0 \end{bmatrix}$, where $\mathcal{G}_0 \in \mathbb{R}^{d' \times d'}$ is a square Gaussian matrix. Thus,

$$\Pr[h_f(x) = h_f(y)] = \Pr\left[h\left(\begin{bmatrix} \tilde{x} \\ 0 \end{bmatrix}\right) = h\left(\begin{bmatrix} \tilde{y} \\ 0 \end{bmatrix}\right)\right],$$

recalling that h_f is the fast cross-polytope hash function and h is the standard version. It then follows that, provided the distance between $\tilde{x}$ and $\tilde{y}$ is close to the distance between x and y, we can apply proposition 3 to control the above probability. We start with a lemma for our chosen JL transform that combines a recent improvement on the *restricted isometry property* (RIP) for partial Hadamard matrices [11] with a reduction from RIP to Johnson-Lindenstrauss transforms in [15]; we defer the proof to the appendix.

▶ **Lemma 7.** *Suppose $\gamma > 0$, $x, y \in S^{d-1}$, $\widetilde{x} = H_S D_b x$, $\widetilde{y} = H_S D_b y$ and $H_S \in \mathbb{R}^{m\times d}$ is such that $m = \mathcal{O}(\gamma \ln^4(d) \ln^4(\ln d))$. Then with probability $1 - \mathcal{O}(d^{-\gamma})$,*

$$\left(1 - \frac{1}{\ln d}\right) \leq \|\widetilde{x}\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right), \tag{7}$$

$$\left(1 - \frac{1}{\ln d}\right) \leq \|\widetilde{y}\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right), \tag{8}$$

$$\left(1 - \frac{1}{\ln d}\right) \|x - y\|_2^2 \leq \|\widetilde{x} - \widetilde{y}\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right) \|x - y\|_2^2 \tag{9}$$

We apply the above lemma with the choice $\gamma = \ln d$ to get that

$$\frac{\|x-y\|_2^2}{(1-\frac{1}{\ln d})} - \frac{5}{\ln d - 1} \leq \left\| \frac{\widetilde{x}}{\|\widetilde{x}\|_2} - \frac{\widetilde{y}}{\|\widetilde{y}\|_2} \right\|_2^2 \leq \frac{\|x-y\|_2^2}{(1+\frac{1}{\ln d})} + \frac{5}{\ln d + 1}. \tag{10}$$

with probability $1 - \mathcal{O}(d^{-\ln d})$. Combining this fact with proposition 3 we get that

$$\Pr[h_f(x) = h_f(y)] = C(d')^{\frac{-\tilde{R}^2}{4-\tilde{R}^2}} \ln^{-1}(d'),$$

where $\tilde{R} = \|\tilde{x} - \tilde{y}\|^2$ (by equation (10)) goes to R as $d \to \infty$, and C is bounded in the dimension. We then apply lemma 6 to see that

$$\begin{aligned}\rho &= \frac{\frac{\tilde{R}^2}{4-\tilde{R}^2} \ln(d') + \ln\ln(d') + C}{\frac{c^2\tilde{R}^2}{4-c^2\tilde{R}^2} \ln(d') + \ln\ln(d') + C} \\ &= \frac{1}{c^2} \frac{4 - c^2R^2}{4 - R^2} + o(1).\end{aligned}$$

We defer the proof of Theorem 4 part (ii-) to the appendix.

3.2 Theorem 5 Proof Outline

We will use the following result (formulated as an analogue to lemma 7) , due to Kane and Nelson, that reduces the amount of randomness required to perform a JL transform.

▶ **Proposition 8.** *(Theorem 13 and Remark 14 in [14]) Suppose $\gamma > 0$, $x, y \in S^{d-1}$. Then, there is a random matrix $Z \in \mathbb{R}^{d\times m}$ with $m = \mathcal{O}(\gamma \ln^3(d))$ and sampled with $\mathcal{O}(\gamma \ln^2(d))$ random bits such that with probability $1 - \mathcal{O}(d^{-\gamma})$,*

$$\left(1 - \frac{1}{\ln d}\right) \leq \|Zx\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right),$$

$$\left(1 - \frac{1}{\ln d}\right) \leq \|Zy\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right),$$

$$\left(1 - \frac{1}{\ln d}\right) \|x - y\|_2^2 \leq \|Z(x-y)\|_2^2 \leq \left(1 + \frac{1}{\ln d}\right) \|x - y\|_2^2$$

Now we want to construct a hash scheme that uses a Gaussian rotation with which to compare our discretized scheme. Define

$$h'_D(x) = \operatorname*{argmin}_{u=\{\pm e_i\}} \left\| \frac{\mathcal{G}'Zx}{\|\mathcal{G}'Zx\|_2} - u \right\|_2, \tag{11}$$

where $\mathcal{G}' \in \mathbb{R}^{m\times m}$ is a standard i.i.d. Gaussian matrix. The following elementary lemma gives us a suitable replacement for each Gaussian in the matrix $\mathcal{G}'$.

▶ **Lemma 9.** *Suppose $g \sim \mathcal{N}(0,1)$. Then, there is a symmetric, discrete random variable X taking 2^b values such that for any $x \in \mathbb{R}$,*

$$\Pr[g \leq x] = \Pr[X \leq x] + \mathcal{O}(2^{-b}) \tag{12}$$

The discretized scheme can now be constructed by

$$h_D(x) = \underset{u=\{\pm e_i\}}{\operatorname{argmin}} \left\| \frac{\widehat{\mathcal{G}}Zx}{\|\widehat{\mathcal{G}}Zx\|_2} - u \right\|_2, \tag{13}$$

where the entries of $\widehat{\mathcal{G}} \in \mathbb{R}^{d' \times m}$ are i.i.d. copies of the random variable X in Lemma 9. Note that each discrete random variable has b bits of randomness, so the hashing scheme has minimial randomness when $d' = m$, thus there are $m \times m \times b + \mathcal{O}(\gamma \ln^2(d)) = \mathcal{O}(\gamma^2 \ln^6(d) b + \gamma \ln^2(d))$ bits of randomness. As we will see, we can choose γ and b to be a power of $\ln(d)$ while still achieve the optimal asymptotic ρ. For this we have the following lemma.

▶ **Lemma 10.** *Let $x, y \in \mathbb{R}^d$ be such that $\|x - y\|_2 = R$, $\widetilde{x} = Zx$, and let h, h' be as defined in (13) and (11) respectively with $m = \mathcal{O}(\ln^4(d))$, $b = \log_2(d)$ where $\widetilde{R} = \|\widetilde{x} - \widetilde{y}\|_2$. Then,*

$$\ln(\Pr[h_D(x) = h_D(y)]) = \ln(\Pr[h'_D(x) = h'_D(y)]) + \mathcal{O}_{\widetilde{R}}(1) \tag{14}$$

We defer the proof of lemma 10 to the appendix, but the idea is as follows. We can first write

$$\Pr[h'_D(x) = h'_D(y)] = 2d' \Pr[h'_D(x) = h'_D(y) = e_1].$$

Note that the set $\{h'_D(x) = h'_D(y) = e_1\} = \{(\mathcal{G}'\widetilde{x})_1 \geq |(\mathcal{G}'\widetilde{x})_2|, (\mathcal{G}'\widetilde{y})_1 \geq |(\mathcal{G}'\widetilde{y})_2|\}$, which is the Gaussian measure of a convex polytope, so we can write the above probability as the integral over m intervals of the m-dimensional Gaussian probability distribution. We can then use equation (12) to replace the Gaussian pdf with the discrete Gaussian pdf in each coordinate succesively, and (keeping track of parameters), the lemma follows.

We now run the same argument as in Theorem 4 by setting $\gamma = \ln d$, so combining lemma 10 and proposition 3 applied to $h'_D(x)$, we have that

$$\begin{aligned}
\rho &= \frac{\ln(\Pr[h_D(x) = h_D(y)])}{\ln(\Pr[h_D(cx) = h_D(cy)])} \\
&= \frac{\ln(\Pr[h'_D(x) = h'(y)]) + \mathcal{O}_{\widetilde{R}}(1)}{\ln(\Pr[h'_D(cx) = h'(cy)]) + \mathcal{O}_{\widetilde{R}}(1)} \\
&= \frac{\frac{R_+^2}{4 - R_+^2} \ln(d') + \ln\ln(d') + C + \mathcal{O}_{\widetilde{R}}(1)}{\frac{c^2 R_-^2}{4 - c^2 R_-^2} \ln(d') + \ln\ln(d') + C + \mathcal{O}_{\widetilde{R}}(1)} \\
&= \frac{\frac{R_+^2}{4 - R_+^2} \ln(d') + \ln\ln(d') + C}{\frac{c^2 R_-^2}{4 - c^2 R_-^2} \ln(d') + \ln\ln(d') + C} \\
&= \frac{1}{c^2} \frac{4 - c^2 R^2}{4 - R^2} + o(1), \text{ by lemma 6.}
\end{aligned}$$

Finally, by our choice of γ and b in the above lemma, we know that there are $\mathcal{O}(\ln^9(d))$ bits of randomness.

4 Open Problems

Although we achieve a logarithmic number of bits of randomness in Theorem 5, there is no reason to believe this is optimal among all hash families. More generally, given a particular rate of convergence to the optimal asymptotic sensitivity we would like to know the minimal number of required bits of randomness. Note that by the result in [20], for each dimension d, $c > 0$, and $q > 0$, there is some distance $R > 0$ such that the sensitivity parameter $\rho \geq \frac{1}{c^2} - \mathcal{O}_q\left(\frac{1}{\ln d}\right)$. In light of this result, we would like to know, for a given rate of convergence, whether it gets close to the lower bound $\frac{1}{c^2}$ for all sequences of distances $R = R(d)$. Note that this condition holds for cross-polytope lsh with $f(d) = \mathcal{O}\left(\frac{1}{\ln d}\right)$.

▶ **Problem 11.** *Given a rate of convergence $f(d)$ such that $\lim_{d\to\infty} f(d) = 0$, find the minimal number of bits $\mathcal{O}_f(d)$ such that any hash family $\mathcal{H}$ over the sphere S^{d-1} with $\mathcal{O}_f(d)$ bits of randomness satisfies $\rho = \frac{1}{c^2} + f(d)$ for all sequences $R = R(d)$.*

A more practical question is, given a rate of convergence for ρ, what is the fastest one could compute a hash family achieving this rate.

▶ **Problem 12.** *Given a rate of convergence $f(d)$ as in Problem 11, find the hash family $\mathcal{H}$ over S^{d-1} such that $\rho = \frac{1}{c^2} + f(d)$ for all sequences $R = R(d)$, that also has the fastest hash computations.*

It would be natural to extend our theoretical analysis to the case of hashing a collection of n points simultaneously. In this setting, the embedding dimension of the JL matrix would inherit an additive factor depending on $\ln(n)$. Inspired by the construction in [17] which first sparsifies the data then exploits the restricted isometry property which applies uniformly over all sparse vectors, we can aim for a construction that doesn't depend on the number of data points.

5 Numerical Experiments

To illustrate our theoretical results in the low dimensional case, we ran Monte Carlo simulations to compare the collision probabilities for regular cross-polytope LSH as well as the fast and discrete versions for various values of the original and lifted dimension. We refer to [4] for an in depth comparison of run times for cross-polytope LSH and other popular hashing schemes.

The experiments were run with $N = 20000$ trials. For each trial, two points were fixed at the given distance threshold on the unit sphere (in the plane given by the first two coordinates) and then rotated uniformly at random on the sphere. The discretized scheme used 10 bits of randomness for each entry. The fast, discrete, and regular cross-polytope LSH schemes exhibit similar collision probabilities for small distances, with fast/discrete cross-polytope having marginally higher collision probabilities for larger distances. It is clear that as the lifted dimension decreases, the fast and discrete versions have higher collision probabilities at further distances, which decreases the sensitivity of those schemes.

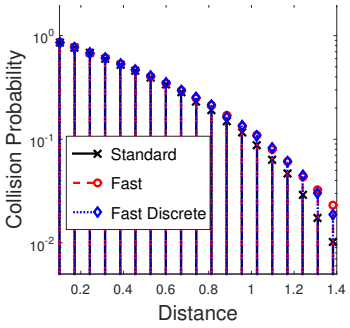


Figure 1 $d = 128$, $d' = 128$.

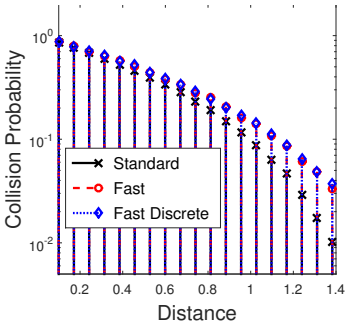


Figure 2 $d = 128$, $d' = 64$.

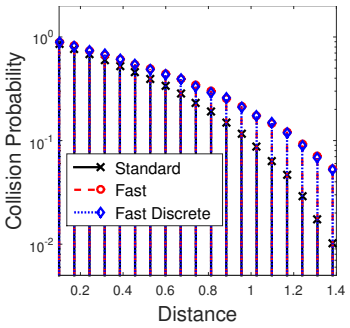


Figure 3 $d = 128$, $d' = 32$.

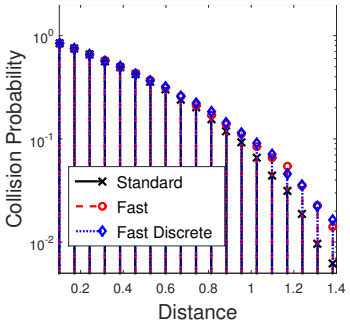


Figure 4 $d = 256$, $d' = 256$.

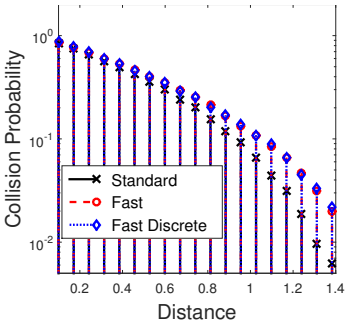


Figure 5 $d = 256$, $d' = 128$.

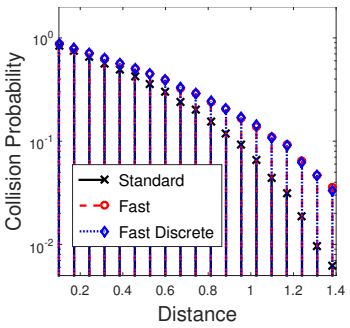


Figure 6 $d = 256$, $d' = 64$.

The following figures illustrate the rate of convergence to the optimal collision probability as $d \to \infty$, as well as various lines that illustrate the optimal rate of convergence $C \setminus \ln(d)$, where C varies for illustrative purposes. The experiments were run with varying distances and clearly show the same rate of convergence for the collision probability between the standard and fast cross-polytope schemes. We note that at low dimensions, the schemes behave even more similarly because the embedded dimension is much closer to the original dimension in this case.

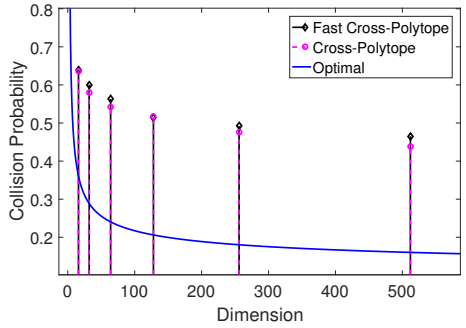


Figure 7 $R = 0.4$.

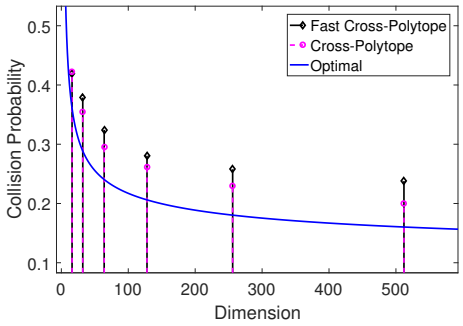


Figure 8 $R = 0.7$.

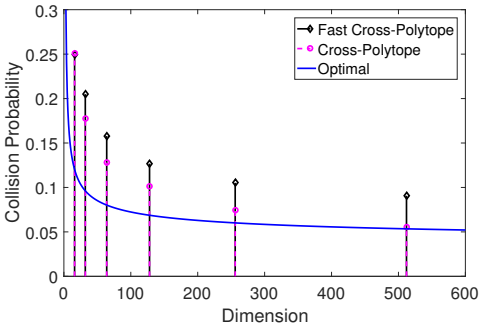


Figure 9 $R = 1$.

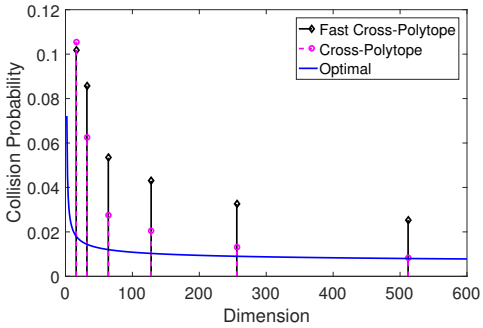


Figure 10 $R = 1.3$.

References

1 Nir Ailon and Bernard Chazelle. The fast Johnson-Lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput.*, 39(1):302–322, May 2009. doi:10.1137/060673096.

2 Noga Alon. Problems and results in extremal combinatorics. *Discrete Mathematics*, 273:31–53, 2003.

3 Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '06, pages 459–468, Washington, DC, USA, 2006. IEEE Computer Society. doi:10.1109/FOCS.2006.49.

4 Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, and Ludwig Schmidt. Practical and optimal lsh for angular distance. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 1225–1233,

Cambridge, MA, USA, 2015. MIT Press. URL: http://dl.acm.org/citation.cfm?id=2969239.2969376.

5 Alexandr Andoni, Piotr Indyk, Huy L Nguyen, and Ilya Razenshteyn. Beyond Locality-Sensitive Hashing. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1018–1028. SIAM, 2014.

6 Alexandr Andoni and Ilya Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, STOC '15, pages 793–801, New York, NY, USA, 2015. ACM. doi:10.1145/2746539.2746553.

7 Alexandr Andoni and Ilya Razenshteyn. Tight Lower Bounds for Data-Dependent Locality-Sensitive Hashing. *ArXiv e-prints*, July 2015. arXiv:1507.04299.

8 Anja Becker and Thijs Laarhoven. Efficient (ideal) lattice sieving using cross-polytope lsh. Cryptology ePrint Archive, Report 2015/823, 2015. URL: http://eprint.iacr.org/.

9 Anirban Dasgupta, Ravi Kumar, and Tamás Sarlós. Fast locality-sensitive hashing. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1073–1081. ACM, 2011.

10 Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab Mirrokni. Locality sensitive hashing scheme based on p-stable distributions. In *Proceedings of the tmentieth annual Symposium on Computational Geometry. New York*, pages 253–262, 2004.

11 Ishay Haviv and Oded Regev. The restricted isometry property of subsampled fourier matrices. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 288–297, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=2884435.2884457.

12 Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 604–613, New York, NY, USA, 1998. ACM. doi:10.1145/276698.276876.

13 William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26:189–206, 1984.

14 Daniel M. Kane and Jelani Nelson. Sparser johnson-lindenstrauss transforms. *J. ACM*, 61(1):4:1–4:23, January 2014. doi:10.1145/2559902.

15 Felix Krahmer and Rachel Ward. New and improved Johnson-Lindenstrauss embeddings via the restricted isometry property. *SIAM Journal on Mathematical Analysis*, 43(3):1269–1281, 2011.

16 Kasper Green Larsen and Jelani Nelson. The johnson-lindenstrauss lemma is optimal for linear dimensionality reduction. *arXiv preprint arXiv:1411.2404*, 2014.

17 Yue Lin, Rong Jin, Deng Cai, Shuicheng Yan, and Xuelong Li. Compressed hashing. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, pages 446–451, Washington, DC, USA, 2013. IEEE Computer Society. doi:10.1109/CVPR.2013.64.

18 Ting Liu, Andrew W. Moore, Ke Yang, and Alexander G. Gray. An investigation of practical approximate nearest neighbor algorithms. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 825–832. MIT Press, 2005. URL: http://papers.nips.cc/paper/2666-an-investigation-of-practical-approximate-nearest-neighbor-algorithms.pdf.

19 Rajeev Motwani, Assaf Naor, and Rina Panigrahi. Lower bounds on Locality Sensitive Hashing. In *Proceedings of the Twenty-second Annual Symposium on Computa-*

tional Geometry, SCG '06, pages 154–157, New York, NY, USA, 2006. ACM. doi:10.1145/1137856.1137881.

20 Ryan O'Donnell, Yi Wu, and Yuan Zhou. Optimal lower bounds for Locality-Sensitive Hashing (except when Q is tiny). *ACM Trans. Comput. Theory*, 6(1):5:1–5:13, March 2014. doi:10.1145/2578221.

21 Andrei Osipov. *A Randomized Approximate Nearest Neighbors Algorithm*. PhD thesis, Yale University, New Haven, CT, USA, 2011. AAI3467911.

22 Kengo Terasawa and Yuzuru Tanaka. Spherical LSH for approximate nearest neighbor search on unit hypersphere. In *Algorithms and Data Structures*, pages 27–38. Springer, 2007.

23 Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, VLDB '98, pages 194–205, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

6 Appendix

6.1 Proof of Theorem 4 Part (ii-)

Let $\rho_{R,c}$ be the exponent for standard cross-polytope lsh in dimension d', and $\rho_{R,c}^{fast}$ be the exponent for fast cross-polytope lsh lifted to dimension d'. Suppose that

$$\rho_{R,c} - \frac{1}{c^2}\frac{4-c^2R^2}{4-R^2} \leq C(R,c)F(d'),$$

where $F(d') \to 0$ as $d' \to \infty$ and $C(r,c)$ is constant in the dimension d'.
Assume that $H_sD_b : \mathbb{R}^d \to \mathbb{R}^m$ is a δ-isometry on $x-y$, i.e.

$$||x-y||_2^2 \leq R^2 \implies ||\tilde{x}-\tilde{y}||_2^2 \leq (1+\delta)R^2 \tag{15}$$

$$||x-y||_2^2 \geq c^2R^2 \implies ||\tilde{x}-\tilde{y}||_2^2 \geq (1-\delta)c^2R^2. \tag{16}$$

The next observation is that $h_f(x)$ applies the standard cross-polytope lsh scheme on H_sD_bx, so conditioned on H_sD_bx being a δ-isometry, we can analyze the fast scheme in terms of the standard scheme as follows:

$$\rho_{R,c}^{fast} \leq \rho_{R',c'},$$

where $R' = R\sqrt{1+\delta}$, $c' = \sqrt{\frac{1-\delta}{1+\delta}}c$. Now, we can say

$$\begin{aligned}\rho_{R,c}^{fast} - \frac{1}{c^2}\frac{4-c^2R^2}{4-R^2} &\leq [\rho_{R,c}^{fast} - \rho_{R',c'}] + \left[\rho_{R',c'} - \frac{1}{(c')^2}\frac{4-(c')^2(r')^2}{4-(R')^2}\right] \\ &\quad + \left[\frac{1}{(c')^2}\frac{4-(c')^2(R')^2}{4-(R')^2} - \frac{1}{c^2}\frac{4-c^2R^2}{4-R^2}\right] \\ &\leq C(R',c')F(d) + \left[\frac{1}{(c')^2}\frac{4-(c')^2(R')^2}{4-(R')^2} - \frac{1}{c^2}\frac{4-c^2R^2}{4-R^2}\right].\end{aligned}$$

The difference in the last equation can be bounded as

$$\begin{aligned}
&\frac{1}{(c')^2}\frac{4-(c')^2(R')^2}{4-(R')^2} - \frac{1}{c^2}\frac{4-c^2R^2}{4-R^2} = \left(\frac{1+\delta}{c^2(1-\delta)}\right)\frac{4-(1-\delta)c^2R^2}{4-(1-\delta)R^2} - \frac{1}{c^2}\frac{4-c^2R^2}{4-R^2} \\
&\leq \frac{(1+\delta)(4-(1-\delta)c^2R^2)(4-R^2) - (4-c^2R^2)(1-\delta)(4-(1-\delta)R^2)}{\frac{c^2}{2}(4-R^2)^2} \\
&= \delta\mathcal{O}(R,c) + \frac{(1+\delta)(4-c^2R^2)(4-R^2) - (1-\delta)(4-c^2R^2)(4-R^2)}{\frac{c^2}{2}(4-R^2)^2} \\
&= \delta D(R,c),
\end{aligned}$$

so it follows that $\rho^{fast}_{R,c} - \frac{1}{c^2}\frac{4-c^2R^2}{4-R^2} \leq \delta D(R,c) + C(R',c')F(d')$ conditioned on the fact that H_sD_b is a δ-isometry on $x-y$. Note that for d' large enough, $C(R',c')$ is bounded above by a constant independent of the dimension. We can make the choice $\delta = \frac{1}{\ln(d)}$, so that the isometry condition holds with probability $1-\mathcal{O}(d^{-\ln d})$, so if ρ is the true exponent without conditioning, we get that

$$\begin{aligned}
\rho &\leq \frac{p_1}{p_2 + C\ln(1-d^{-\ln d})} \\
&\leq \frac{p_1}{p_2 - Cd^{-\ln d}} \\
&\leq \frac{p_1}{p_2}(1 + Cd^{-\ln d}/p_1),
\end{aligned}$$

where $C>0$ is an constant that changes by line but is independent of the dimension. From this expression it is easy to see that the error term decays at least like $1/\ln d'$ (recall that $d' \leq d$).
Finally, provided $F(d')$ decays as fast as than $\frac{1}{\ln(d')}$, the result will hold. This follows from Theorem 1 in [4].

6.2 Proof of Lemma 6

We know that for any $\epsilon > 0$ and d large enough, $m_d(b) \geq b - \epsilon$, so that

$$\begin{aligned}
\lim_{d\to\infty}\frac{g(d)}{m_d(b)f(d)+g(d)} &\leq \lim_{d\to\infty}\frac{g(d)}{(b-\epsilon)f(d)+g(d)} \\
&= \lim_{d\to\infty}\frac{1}{(b-\epsilon)\frac{f(d)}{g(d)}+1} = 0,
\end{aligned}$$

and by positivity the inequality is an equality. This implies that

$$\lim_{d\to\infty}\frac{m_d(a)f(d)+g(d)}{m_d(b)f(d)+g(d)} = \lim_{d\to\infty}\frac{m_d(a)f(d)}{m_d(b)f(d)+g(d)}.$$

The same argument on the reciprocal shows that

$$\lim_{d\to\infty}\frac{m_d(a)f(d)}{m_d(b)f(d)+g(d)} = \lim_{d\to\infty}\frac{m_d(a)f(d)}{m_d(b)f(d)} = \frac{a}{b}$$

6.3 Proof of Lemma 7

Define the event

$$E_{v,\delta} := \{v \in \mathbb{R}^n : (1-\delta)\|v\|_2 \leq \|\widetilde{v}\|_2 \leq (1+\delta)\|v\|_2\}.$$

sCombining Theorem 4.5 of [11] and Theorem 3.1 of [15], we know that for any $\eta \in (0,1)$, any $s \geq 40\ln(12/\eta)$, some $C_0 > 0$, and provided $m = \mathcal{O}(\delta^{-2}\ln^2(1/\delta)s\ln^2(s/\delta)\ln(d))$,

$$\Pr[E_{x,\delta} \cap E_{y,\delta} \cap E_{x-y,\delta}] \geq (1-\eta)(1-2^{-C_0\ln(d)\ln(s/\delta)})$$

Setting $\delta = 1/\ln(d)$, $\eta = d^{-\gamma}$, $s = 40C\ln(12d)$, we get

$$\Pr[E_{x,\delta} \cap E_{y,\delta} \cap E_{x-y,\delta}] \geq (1-d^{-\gamma})(1-2^{-C_0\ln(d)\ln(40\gamma\ln(12d)\ln(d))}),$$

and the lemma follows.

6.4 Proof of Lemma 10

Note that since the entries of $\widehat{\mathcal{G}}\widetilde{x}$ are symmetric and i.i.d., the probability of hashing to one value is equal for all hash values, so we get

$$\begin{aligned}\Pr[h_D(x) = h_D(y)] &= 2d'\Pr[h_D(x) = h_D(y) = e_1] \\ &= 2d'\Pr[\cap_{j=2}^{d'}(\widehat{\mathcal{G}}\widetilde{x})_1 \geq |(\widehat{\mathcal{G}}\widetilde{x})_j|, (\widehat{\mathcal{G}}\widetilde{y})_1 \geq |(\widehat{\mathcal{G}}\widetilde{y})_j|] \\ &= 2d'\mathbb{E}_{(\widehat{\mathcal{G}}\widetilde{x})_1,(\widehat{\mathcal{G}}\widetilde{y})_1}(\Pr[(\widehat{\mathcal{G}}\widetilde{x})_1 \geq |(\widehat{\mathcal{G}}\widetilde{x})_2|, (\widehat{\mathcal{G}}\widetilde{y})_1 \geq |(\widehat{\mathcal{G}}\widetilde{y})_2|]^{d'-1}). \qquad (17)\end{aligned}$$

Our goal is to bound the probability $\Pr[(\widehat{\mathcal{G}}\widetilde{x})_1 \geq |(\widehat{\mathcal{G}}\widetilde{x})_2|, (\widehat{\mathcal{G}}\widetilde{y})_1 \geq |(\widehat{\mathcal{G}}\widetilde{y})_2|]$ in terms of the probability $\Pr[(\mathcal{G}'\widetilde{x})_1 \geq |(\mathcal{G}'\widetilde{x})_2|, (\mathcal{G}'\widetilde{y})_1 \geq |(\mathcal{G}'\widetilde{y})_2|]$. Define $E_{\mathcal{G}'} = \{(\mathcal{G}'\widetilde{x})_1 \geq |(\mathcal{G}'\widetilde{x})_2|, (\mathcal{G}'\widetilde{y})_1 \geq |(\mathcal{G}'\widetilde{y})_2|\}$ and similarly for $\widehat{\mathcal{G}}$. Since $E_{\mathcal{G}'}$ is a convex polytope, we can write

$$\Pr[E_{\mathcal{G}'}] = \int_{I_1}\int_{I_2(x_1)}\dots\int_{I_m(x_1,x_2,\dots,x_{m-1})}\frac{1}{(2\pi)^m}e^{-(x_1^2+\dots+x_m^2)/2}dx_m\dots dx_1,$$

where each $I_j(x_1,\dots,x_j)$ is a (possibly unbounded) interval. By construction of X,

$$\int_{I_j(x_1,\dots,x_j)}\frac{1}{2\pi}e^{-x_{j+1}^2/2}dx_{j+1} = \int_{I_j(x_1,\dots,x_j)}p_X(x_{j+1})dx_{j+1} + \mathcal{O}(2^{-b})$$

where $p_X(x)$ is the pdf of X. This implies that

$$\begin{aligned}&\Pr[E_{\mathcal{G}'}] \\ &= \int_{I_1}\int_{I_2(x_1)}\dots\int_{I_m(x_1,\dots,x_{m-1})}\frac{1}{(2\pi)^{m-1}}e^{-(x_1^2+\dots+x_{m-1}^2)/2}p_X(x_m)dx_m\dots dx_1 + \mathcal{O}(2^{-b}) \\ &\dots = \int_{I_1}\int_{I_2(x_1)}\dots\int_{I_m(x_1,\dots,x_{m-1})}p_X(x_1)\dots p_X(x_m)dx_m\dots dx_1 + \mathcal{O}(m2^{-b}) \\ &= \Pr[E_{\widehat{\mathcal{G}}}] + \mathcal{O}(m2^{-b}).\end{aligned}$$

Plugging this into (17), we get

$$\begin{aligned}\Pr[h_D(x) = h_D(y)] &= 2d'\mathbb{E}_{(\widehat{\mathcal{G}}\widetilde{x})_1,(\widehat{\mathcal{G}}\widetilde{y})_1}(\Pr[E_{\mathcal{G}'}] + \mathcal{O}(m2^{-b})))^{d'-1} \\ &= 2d'\mathbb{E}_{(\widehat{\mathcal{G}}\widetilde{x})_1,(\widehat{\mathcal{G}}\widetilde{y})_1}\left[\sum_{k=1}^{d'-1}\binom{d'-1}{k}\Pr[E_{\mathcal{G}'}]^k(\mathcal{O}(m2^{-b}))^{d'-1-k}\right].\end{aligned}$$

We now make the choice $m = C\ln^4(d)$, $b = \log_2(d)\ln(d)$, so that the above summation becomes

$$\begin{aligned}&\sum_{k=1}^{d'-1}\binom{d'-1}{k}\Pr[E_{\mathcal{G}'}]^{d'-1-k}(C\ln^4(d)d^{-\ln(d)})^k \\ &\qquad = \sum_{k=1}^{d'-1}\binom{d'-1}{k}\Pr[E_{\mathcal{G}'}]^{d'-1-k}(C\ln^4(d)d^{-\ln(d)})^k\end{aligned}$$

This first term in the summation is the main term $\Pr[E_{\mathcal{G}'}]^{d'-1}$ and the other terms can be bounded using Sterling's approximation as follows,

$$\binom{d'-1}{k}\Pr[E_{\mathcal{G}'}]^{d'-1-k}(C\ln^4(d)d^{-\ln(d)})^k \leq \left(\frac{d'e}{k}\right)^k (C\ln^4(d)d^{-\ln(d)})^k.$$

For $k \geq 1$ this is certainly bounded by $\mathcal{O}(d^{-\ln(d)+1})$, and we have

$$\begin{aligned}\sum_{k=1}^{d'-1}\binom{d'-1}{k}&\Pr[E_{\mathcal{G}'}]^{d'-1-k}(C\ln^4(d)d^{-\ln(d)})^k \\ &= \Pr[E_{\mathcal{G}'}]^{d'-1} + \mathcal{O}(d^{-\ln(d)+2})\end{aligned}$$

We note that the last asymptotic approximation is very rough but sufficient for our purposes. This means that

$$\Pr[h_D(x) = h_D(y)] = 2d'\mathbb{E}_{(\widehat{\mathcal{G}\widetilde{x}})_1,(\widehat{\mathcal{G}\widetilde{y}})_1}(\Pr[E_{\mathcal{G}'}]^{d'-1}) + \mathcal{O}(md^{-\ln(d)+2}). \tag{18}$$

Using the same technique as above where we replace the Gaussian density function with $P_X(x)$, we have

$$\begin{aligned}\Pr[h'_D(x) = h'_D(y)] &= 2d'\mathbb{E}_{(\mathcal{G}'\widetilde{x})_1,(\mathcal{G}'\widetilde{y})_1}(\Pr[E_{\mathcal{G}'}]^{d'-1}) \\ &= 2d'\mathbb{E}_{(\widehat{\mathcal{G}\widetilde{x}})_1,(\widehat{\mathcal{G}\widetilde{y}})_2}(\Pr[E_{\mathcal{G}'}] + \mathcal{O}(m2^{-b}))^{d'-1} \\ &= 2d'\mathbb{E}_{(\widehat{\mathcal{G}\widetilde{x}})_1,(\widehat{\mathcal{G}\widetilde{y}})_2}(\Pr[E_{\mathcal{G}'}]^{d'-1}) + \mathcal{O}(md^{-\ln(d)+2})\end{aligned}$$

Finally, plugging this into (18), we get

$$\begin{aligned}\Pr[h_D(x) = h_D(y)] &= \Pr[h'_D(x) = h'_D(y)] + \mathcal{O}(md^{-\ln(d)+2}) \\ &= \Pr[h'_D(x) = h'_D(y)] + \mathcal{O}(d^{-\ln(d)+3}).\end{aligned}$$

Now, we know that by Theorem 3, $\ln(\Pr[h_D(x) = h_D(y)]) = -\frac{\widetilde{R}^2}{4-\widetilde{R}^2}\ln(d') + \mathcal{O}_{\widetilde{R}}(\ln(\ln d'))$, so provided d is large enough that $\ln(d) - 2 > \frac{\widetilde{R}^2}{4-\widetilde{R}^2}$, the lemma follows.

The Distortion of Locality Sensitive Hashing

Flavio Chierichetti*[1], Ravi Kumar[2], Alessandro Panconesi*[3], and Erisa Terolli*[4]

1 **Dipartimento di Informatica, Sapienza University of Rome, Rome, Italy**
`flavio@di.uniroma1.it`
2 **Google, Mountain View, USA**
`ravi.k53@gmail.com`
3 **Dipartimento di Informatica, Sapienza University of Rome, Rome, Italy**
`ale@di.uniroma1.it`
4 **Dipartimento di Informatica, Sapienza University of Rome, Rome, Italy**
`terolli@di.uniroma1.it`

Abstract

Given a pairwise similarity notion between objects, locality sensitive hashing (LSH) aims to construct a hash function family over the universe of objects such that the probability two objects hash to the same value is their similarity. LSH is a powerful algorithmic tool for large-scale applications and much work has been done to understand LSHable similarities, i.e., similarities that admit an LSH.

In this paper we focus on similarities that are provably non-LSHable and propose a notion of distortion to capture the approximation of such a similarity by a similarity that is LSHable. We consider several well-known non-LSHable similarities and show tight upper and lower bounds on their distortion.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases Locality sensitive hashing, Distortion, Similarity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.54

1 Introduction

The notion of similarity finds its use in a large variety of fields above and beyond computer science. Often, the notion is tailored to the actual domain and the application for which it is intended. Locality sensitive hashing (henceforth LSH) is a powerful algorithmic paradigm for computing similarities between data objects in an efficient way. Informally, an LSH scheme for a similarity is a probability distribution over a family of hash functions such that the probability the hash values of two objects agree is precisely the similarity between them. In many applications, computing similar objects (i.e., finding nearest neighbors) can be computationally very demanding and LSH offers an elegant and cost-effective alternative.

Intuitively, large objects can be represented compactly and yet accurately from the point of view of similarity, thanks to LSH. Thus, the similarity between two objects can be quickly estimated by picking a few random hash functions from the family and estimating the fraction of times the hash functions agree on the two objects. This paradigm has been very successful in a variety of applications dealing with large volumes of data, from near-duplicate estimation in text corpora to nearest-neighbor search in a multitude of domains.

* These authors were partially supported by a Google Focused Research Award, by the ERC Starting Grant DMAP 680153, and by the SIR Grant RBSI14Q743.


© Flavio Chierichetti, Ravi Kumar, Alessandro Panconesi, and Erisa Terolli;
licensed under Creative Commons License CC-BY

Given its success and importance[1], researchers have looked for LSH schemes for more and more similarities. Thus a natural question arises: which similarities admit an LSH scheme? In [12] Charikar introduced two necessary criteria (the former weaker than the latter) for a similarity S to admit an LSH:

(T1) $1 - S$ must be a metric;

(T2) $1 - S$ must be isometrically embeddable in ℓ_1.

These two tests can be used to rule out the existence of LSH schemes for various similarities, for instance, the Sørensen–Dice and Sokal–Sneath similarities (see Table 1 or [15] for definitions).

This brings us to a very natural question, and the one we address in this paper: *if a similarity S does not admit an LSH scheme, then how well can it be approximated by another similarity S' that admits an LSH?*

Locality sensitive distortion. The two criteria (T1) and (T2) are one of the many points of contact between LSH schemes and the theory of embeddability in metric spaces, where the natural notion of "closeness" is distortion. We say that a similarity S has a *distortion* not larger than δ if there is a similarity S' defined on the same universe that admits an LSH and such that

$$\frac{S}{\delta} \leq S' \leq S.$$

The distortion is 1 if and only if S admits an LSH.

In this paper we begin a systematic investigation of the notion of distortion for LSH schemes and prove optimal distortion bounds for several well-known and widely used similarities such as cosine, Simpson, Braun–Blanquet (also known as "all-confidence"), Sørensen–Dice and several others (see Table 1). We obtain our lower bounds by introducing two new combinatorial tools dubbed the *center method* and the *k-sets method*. In nearly all cases, we also exhibit matching distortion upper bounds by explicitly constructing an LSH. As concrete examples, we show that the distortion of cosine similarity is $\Theta(\sqrt{n})$ and that of Braun–Blanquet and Sørensen–Dice similarities is two (the full picture is given in Table 1).

Our framework also greatly expands the outreach of the tests (T1) and (T2). We demonstrate its applicability by means of a few notable examples, in particular the Braun–Blanquet similarity whose distortion is proven to be exactly two. This similarity is particularly noteworthy because not only it passes test (T1) but also (T2). To show this we prove that this similarity is embeddable isometrically in ℓ_1, a result that may be of independent interest. Besides the two general methods discussed, we also provide ad hoc distortion bounds for Sokal–Sneath 1 and Forbes similarities.

Of the two methods introduced in our work, the center method is easier to apply than the k-sets method. The former is applicable to many instances of similarity but the latter is unavoidable in the following sense. Braun–Blanquet similarity not only, as remarked, passes (T1) and (T2), but also the test provided by the center method. However, the more powerful k-sets method can instead be used to show a distortion bound of two. Other similarities to which the k-sets method applies are Sørensen–Dice and the family SORENSEN_γ.

[1] The 2012 Paris Kanellakis Theory and Practice Award was given to Broder, Charikar, and Indyk for their work on LSH.

Table 1 A list of similarities and of their lower and upper distortion bounds. The value n refers to the cardinality of the ground set or to the number of dimensions.

Name	$S(X,Y)$ $X \neq Y$	Distortion LB	Distortion UB
Jaccard	$\frac{\lvert X\cap Y\rvert}{\lvert X\cap Y\rvert+\lvert X\triangle Y\rvert}$	1	1 (Shingles [8])
Hamming	$\frac{\lvert X\cap Y\rvert+\lvert\overline{X\cup Y}\rvert}{\lvert X\cap Y\rvert+\lvert\overline{X\cup Y}\rvert+\lvert X\triangle Y\rvert}$	1	1 (folklore)
Anderberg	$\frac{\lvert X\cap Y\rvert}{\lvert X\cap Y\rvert+2\lvert X\triangle Y\rvert}$	1	1 (RSS [13])
Rogers–Tanimoto	$\frac{\lvert X\cap Y\rvert+\lvert\overline{X\cup Y}\rvert}{\lvert X\cap Y\rvert+\lvert\overline{X\cup Y}\rvert+2\lvert X\triangle Y\rvert}$	1	1 (RSS [13])
Cosine	$\frac{X\cdot Y}{\ell_2(X)\cdot\ell_2(Y)}$	$\sqrt{n}$ (Theorem 6)	$3\sqrt{n}$ (Theorem 7)
Simpson	$\frac{\lvert X\cap Y\rvert}{\min\{\lvert X\rvert,\lvert Y\rvert\}}$	n (Theorem 5)	n (Shingles [8])
Braun–Blanquet	$\frac{\lvert X\cap Y\rvert}{\max\{\lvert X\rvert,\lvert Y\rvert\}}$	2 (Theorem 16)	2 (Shingles [8])
Sørensen–Dice	$\frac{\lvert X\cap Y\rvert}{\lvert X\cap Y\rvert+1/2\lvert X\triangle Y\rvert}$	2 (Theorem 5)	2 (Shingles [8])
Sokal–Sneath 1	$\frac{\lvert X\cap Y\rvert+\lvert\overline{X\cup Y}\rvert}{\lvert X\cap Y\rvert+\lvert\overline{X\cup Y}\rvert+1/2\lvert X\triangle Y\rvert}$	4/3 (Theorem 8)	2 (RSS [13])
Forbes	$\frac{n\lvert X\cap Y\rvert}{\lvert X\rvert\lvert Y\rvert}$	n (Theorem 18)	n (Theorem 18)
SORENSEN$_\gamma$	$\frac{\lvert X\cap Y\rvert}{\lvert X\cap Y\rvert+\gamma\lvert X\triangle Y\rvert}$	$\max(1,1/\gamma)$ (Theorem 5)	$\max(1,1/\gamma)$ (Shingles [8], RSS [13])
SOKAL-SNEATH$_\gamma$	$\frac{\lvert X\cap Y\rvert+\lvert\overline{X\cup Y}\rvert}{\lvert X\cap Y\rvert+\lvert\overline{X\cup Y}\rvert+\gamma\lvert X\triangle Y\rvert}$	$\max(1,2/(1+\gamma))$ (Theorem 8)	$\max(1,1/\gamma)$ (RSS [13])

Upper bounds: worst-case vs. practice. The main motivation behind our work is to extend the range of applicability of LSH as far as possible and our concept of distortion should be understood in these terms. For instance, even if a similarity is shown not to admit an LSH scheme it might be possible to approximate it efficiently by means of LSH schemes of other similarities that are close to it. Our results show that some cases, such as cosine, are a forlorn hope (since the distortion is not a constant), but in other instances, such as Sørensen–Dice and Braun–Blanquet, our bounds give reasons to be optimistic. As a first "proof of concept" of the notion of distortion we performed a series of experiments with real-world text corpora. The results are encouraging, for they show that the distortion of real data sets is smaller than the worst case. In our tests the average distortion turned out to be approximately 1.4 as opposed to the worst-case bound of two.

In the same vein we also investigate experimentally for the first time the effectiveness of two recent LSH schemes for Anderberg and Rogers–Tanimoto similarities. Until the work in [13] it was not known whether these similarities admitted LSH schemes. That paper shows that they do, in a somewhat peculiar way—strictly speaking they might need exponentially many bits (albeit with low probability)! In this paper we report on experiments with real text corpora that show that in practice these schemes are quite efficient.

2 Related Work

LSH was formally developed over a series of papers [8, 9, 25, 26]. Broder et al. [8, 9] showed that min-wise independent permutations form an LSH for the Jaccard similarity. Indyk and Motwani [25] introduced sampling hash as an LSH scheme for the Hamming similarity. Pursuing the work of characterizing similarities that admit an LSH, Charikar [12] introduced (T1) and (T2) as necessary criteria. Chierichetti and Kumar [13] proposed the concept of LSH-preserving functions, which are probability generating functions that preserve the LSH property of a similarity. From applications point of view, LSH has been widely used for solving the approximate or exact near-neighbor search [2] and similarity search [22, 30, 38] in high dimensional spaces. For a detailed bibliography on LSH, including pointers to implementations, see Alex Andoni's LSH page (`www.mit.edu/~andoni/LSH/`) and the surveys of Andoni and Indyk [3] and Wang et al. [43].

Similarities are extensively used in various areas of computer science. Hamming similarity, for instance, is widely used in information theory [5, 6, 18]. Areas like data mining and data management have seen the usage of Anderberg similarity [1], Cosine similarity [10, 37], and Sokal–Sneath [40] similarity. Cosine similarity is also ubiquitous in information retrieval [21, 32, 36, 46] and bioinformatics [11] whereas Sokal–Sneath is used in image processing [4]. We should note here that the success of similarity algorithms/functions is not limited only within computer science. For instance, Sørensen–Dice is commonly used in ecology [16, 28, 29], phytosociology [27, 42], plant taxonomy [44], biology [39] and even in lexicography [35]. Biology has also seen the usage of Sokal–Sneath [41, 45], mentioned above. Other interesting examples are Simpson similarity used in microscopy [31] and biology [17], Braun–Blanquet in phytosociology [7] and ecology [33], and Rogers–Tanimoto used in taxonomy [34].

The notion of distortion is studied in various areas of computer science and mathematics, especially in metric embedding problems. Here, we are given a source metric space (X, d), and a target metric space (X', d'), and we wish to find a map $f : X \to X'$ from points in X to points in X' that minimizes the distortion

$$\max_{\{a,b\}\in\binom{X}{2}} \max\left(\frac{d(a,b)}{d'(f(a),f(b))}, \frac{d'(f(a),f(b))}{d(a,b)}\right).$$

Problems of this form have been studied for many source and target metric spaces (cf. [24]). Examples include embeddings into the Euclidean (ℓ_2) metric, into the ℓ_1 metric, or into tree metrics from shortest-path metrics on graphs or from normed spaces of large dimensionality. Even though the LSH distortion problem seems to resemble distorted metric embedding problems, an important difference is that we want to guarantee a multiplicative approximation to the "similarity" (as opposed to the "distance").

3 Preliminaries

We use the notation 2^A to represent the set of all subsets of a set A. Also, for any set A, $\binom{A}{2}$ is the set of all pairs $\{a, b\}$ such that $a \neq b$ and $a, b \in A$. For a positive integer n, let $[n] = \{1, 2, \ldots, n\}$.

Let $\mathcal{U}$ be a (finite) universe of objects. A symmetric function $S : \mathcal{U} \times \mathcal{U} \to [0, 1]$ such that $S(X, X) = 1$ for all $X \in \mathcal{U}$ is called a *similarity*. See [15] for a rather complete illustration of the different types of similarities that are used in a practical context.

We first define what it means for a similarity to admit a locality sensitive hash (LSH).

▶ **Definition 1** (LSH [12]). An *LSH* for a similarity function $S : \mathcal{U} \times \mathcal{U} \to [0,1]$ is a probability distribution over a set $\mathcal{H}$ of (hash) functions defined on $\mathcal{U}$ such that, for each $X, Y \in \mathcal{U}$, we have

$$\Pr_{h \in \mathcal{H}}[h(X) = h(Y)] = S(X,Y).$$

(See [25] for a somewhat different definition of LSH in the same spirit.) A similarity is *LSHable* if there exists an LSH for it. The basic notion we introduce in this paper is defined next.

▶ **Definition 2** (LSH distortion). The *LSH distortion*, or *distortion*, of a similarity $S : \mathcal{U} \times \mathcal{U} \to [0,1]$ is the minimum[2] $\delta \geq 1$ such that there exists an LSHable similarity $S' : \mathcal{U} \times \mathcal{U} \to [0,1]$ for which

$$\frac{1}{\delta} \cdot S(X,Y) \leq S'(X,Y) \leq S(X,Y) \quad \forall X, Y \in \mathcal{U}.$$

We denote $\mathsf{distortion}(S) = \delta$.

At first blush a more general definition seems possible. One could define the distortion of S as the minimum δ such that there exist an LSHable similarity S' and $\alpha, \beta \geq 1$, with $\alpha\beta = \delta$, such that, for all $X, Y \in \mathcal{U}$,

$$\frac{1}{\alpha} \cdot S(X,Y) \leq S'(X,Y) \leq \beta \cdot S(X,Y).$$

The next lemma however implies that Definition 2 can be adopted without loss of generality.

▶ **Lemma 3.** *Let $S : \mathcal{U} \times \mathcal{U} \to [0,1]$ be an LSHable similarity. Then, for each $\gamma \in [0,1]$, the similarity*

$$S'(X,Y) = \begin{cases} \gamma \cdot S(X,Y) & X \neq Y \\ 1 & X = Y \end{cases}$$

is also LSHable.

Proof. Let $\mathcal{H}$ be the hash function family for S given by Definition 1. We will build a family $\mathcal{H}'$ for S' by bijectively obtaining an h' for each $h \in \mathcal{H}$. To define h', consider the following procedure: with probability γ, let $h'(X) = (0, h(X))$ for each $X \in \mathcal{U}$, while with probability $1 - \gamma$, let $h'(X) = (1, X)$, for each $X \in \mathcal{U}$. Then, for each $X \neq Y$, $\Pr[h'(X) = h'(Y)] = \gamma \cdot S'(X,Y)$. ◀

Now, suppose that for a given similarity S, we have an LSHable similarity S' satisfying $\frac{1}{\alpha} \cdot S(X,Y) \leq S'(X,Y) \leq \beta \cdot S(X,Y)$ with $\alpha\beta = \delta$. By applying Lemma 3 to S' we obtain an LSH for the similarity $S''(X,Y) = \frac{1}{\beta} \cdot S'(X,Y)$ (when $X \neq Y$) which satisfies

$$\frac{1}{\alpha\beta} \cdot S(X,Y) \leq \frac{1}{\beta} \cdot S'(X,Y) = S''(X,Y) \leq S(X,Y).$$

Hence Definition 2 is robust.

[2] A minimum δ exists because it is equal to the solution of a linear program (see, e.g., [14]) of size polynomial in $|\mathcal{U}|$.

Known LSH for set similarities. Set similarities are those similarities whose universe $\mathcal{U}$ satisfies $\mathcal{U} = 2^U$, for some finite ground set U. To give upper bounds on the distortions of various similarities we employ a number of LSH schemes for set similarities proposed in the literature. First and foremost, we employ *shingles* [8, 9], which is an LSH scheme for the Jaccard similarity over sets ($\textsc{jaccard}(X, Y) = |X \cap Y|/|X \cup Y|$), over the universe $\mathcal{U} = 2^U$. To sample a hash function $h \in \mathcal{H}$ from this scheme, one picks a permutation π of the ground set U uniformly at random. Then, $h(X)$, for a set $X \neq \varnothing$, is equal to the element in X with smallest rank in π. (And, $h(\varnothing)$ is identically equal to $\perp$.) A simple calculation shows that $\Pr_{h\in\mathcal{H}}[h(X) = h(Y)] = \frac{|X\cap Y|}{|X\cup Y|}$ if $X \cup Y \neq \varnothing$, and $\Pr_{h\in\mathcal{H}}[h(\varnothing) = h(\varnothing)] = 1$.

We also use a generalization of shingles given in [12] for the weighted Jaccard similarity. Finally, we use some of the LSH schemes given in [13] for the various rational set similarities. We will use these results as black-boxes and hence we will not describe them.

4 The Center Method

In this section we introduce our first lower bound tool for LSH distortion. It will be used to get tight bounds for the distortion of Simpson, and two infinite families of similarities, namely, S_γ and ℓ_p-norm dot product, that contain well-known similarities such as Sørensen–Dice and Cosine as special cases. The main workhorse is given by the next theorem. Roughly, it says that if we can find a set of points in our universe that are mutually far apart, then its "center" is far apart from some point in the set. Later in this section, we will also present matching distortion upper bounds for these similarities.

▶ **Theorem 4.** *Suppose that $S : \mathcal{U} \times \mathcal{U} \to [0, 1]$ is a similarity admitting an LSH such that there exists $\varnothing \neq \mathcal{X} \subseteq \mathcal{U}$, with $S(X, X') = 0$ for each $\{X, X'\} \in \binom{\mathcal{X}}{2}$. Then, for each $Y \in \mathcal{U}$, there exists at least one $X^\star \in \mathcal{X}$ such that $S(X^\star, Y) \leq 1/|\mathcal{X}|$.*

Proof. Let $\mathcal{H}$ be the hash function family for S. Observe that no LSH with a finite distortion can assign a non-zero probability to any of the pairs in $\binom{\mathcal{X}}{2}$, since their pairwise similarities are zero. Therefore each $Y \in \mathcal{U}$ can be hashed to the same value of at most one element of $\mathcal{X}$ for each hash function. In other words, each hash function $h \in \mathcal{H}$ must satisfy $|\{h(X) \mid X \in \mathcal{X}\}| = |\mathcal{X}|$. Therefore,

$$\sum_{X\in\mathcal{X}} S(X, Y) = \Pr[h(Y) \in \{h(X) \mid X \in \mathcal{X}\}] \leq 1.$$

By averaging, it follows that there must exists at least one $X^\star \in \mathcal{X}$ such that $S(X^\star, Y) \leq 1/|\mathcal{X}|$. ◀

We will use this characterization in the following way. For a given similarity, we will find a set $\mathcal{X} \subseteq \mathcal{U}$ of objects that are entirely dissimilar from one another (i.e., all their pairwise similarities are zero) and an additional object $Y \in \mathcal{U} \setminus \mathcal{X}$ (i.e., the *center*) that is more similar than $1/|\mathcal{X}|$ to each of the elements in $\mathcal{X}$. If we can prove a lower bound of $\alpha/|\mathcal{X}|$, $\alpha > 1$, on the similarities $S(Y, X)$ for each $X \in \mathcal{X}$, then we can conclude that the similarity S has to be distorted by at least α to admit an LSH. In the remainder of this section we apply Theorem 4 to a few notable examples.

4.1 Simpson and generalized Sørensen–Dice

Let us begin by recalling the definition of the similarities to be discussed in this section. The Simpson similarity, operating on the subsets of the ground set $[n]$, is defined as

$$\textsc{simpson}(X,Y) = \frac{|X \cap Y|}{\min(|X|,|Y|)},$$

if $|X|, |Y| \geq 1$, as $\textsc{simpson}(X, \varnothing) = 0$ if $|X| \geq 1$ and as $\textsc{simpson}(\varnothing, \varnothing) = 1$. The infinite family $\textsc{sorensen}_\gamma$, for $\gamma > 0$, operating on the subsets of $[n]$, is defined as

$$\textsc{sorensen}_\gamma(X,Y) = \frac{|X \cap Y|}{|X \cap Y| + \gamma |X \triangle Y|},$$

if $|X| + |Y| \geq 1$, and $\textsc{sorensen}_\gamma(\varnothing, \varnothing) = 1$. The $\textsc{sorensen}_\gamma$ family subsumes as special cases several well-known similarities, for instance, Sørensen–Dice ($\gamma = \frac{1}{2}$), Jaccard ($\gamma = 1$), and Anderberg ($\gamma = 2$).

▶ **Theorem 5.** *For a ground set of n elements,*

$$\begin{aligned} \mathsf{distortion}(\textsc{simpson}) &= n, \text{ and} \\ \mathsf{distortion}(\textsc{sorensen}_\gamma) &= \max(1/\gamma, 1) - O(1/n). \end{aligned}$$

Proof. First, we show the lower bound by exhibiting an instance on a ground set of n elements. Let $U = [n]$, $Y = U$, and $\mathcal{X} = \{X_1, \ldots, X_n\}$, where $X_i = \{i\}$ for $i \in [n]$. Observe that, for each $\{X_i, X_j\} \in \binom{\mathcal{X}}{2}$, we have that $\textsc{simpson}(X_i, X_j) = \textsc{sorensen}_\gamma(X_i, X_j) = 0$, while, for each $X_i \in \mathcal{X}$, we have $\textsc{simpson}(X_i, Y) = 1$ and $\textsc{sorensen}_\gamma(X_i, Y) = \frac{1}{\gamma n + (1-\gamma)}$.

By Theorem 4 we know that for every similarity S with an LSH that finitely distorts $\textsc{simpson}$ or $\textsc{sorensen}_\gamma$, there must exist at least one X_i such that $S(X_i, Y) \leq \frac{1}{|\mathcal{X}|} = \frac{1}{n}$. The lower bounds follow.

Next we show matching upper bounds for the distortion. Recall the definition of the Jaccard similarity:

$$\textsc{jaccard}(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}.$$

Broder's shingles [8] and minwise independent permutations [9] are a well-known LSH scheme for Jaccard similarity (see § 2). We use this to prove matching upper bounds for Theorem 5.

Minwise independent permutations form an LSH scheme with distortion n for Simpson similarity since

$$\min(|X|,|Y|) \leq |X \cup Y| \leq n \cdot \min(|X|,|Y|),$$

as long as $|X|, |Y| \geq 1$. They also provide a distortion of $1/\gamma$ for $\textsc{sorensen}_\gamma$, for every $\gamma \in (0, 1]$ since

$$\gamma |X \cup Y| \leq |X \cap Y| + \gamma |X \triangle Y| \leq |X \cup Y|.$$

Finally, recall that a result in [13] proves that the similarity H_γ admits an LSH scheme as long as $\gamma \geq 1$. ◀

Figure 1 plots the minimum distortion of $\textsc{sorensen}_\gamma$, as γ varies.

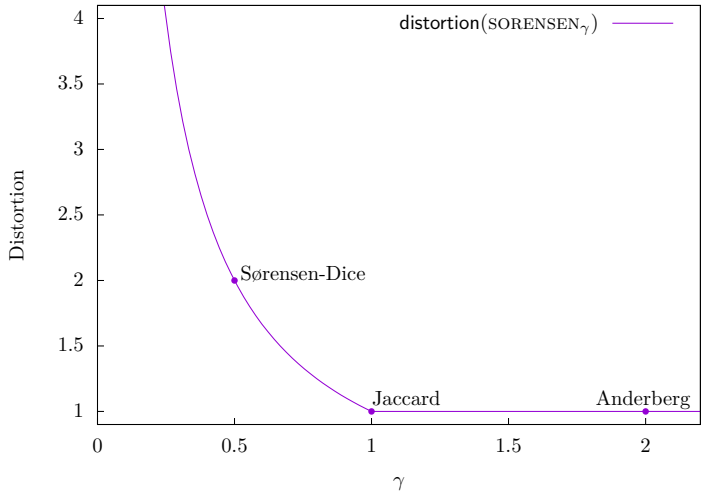


Figure 1 The minimum distortion of SORENSEN$_\gamma$.

4.2 Cosine and unit ℓ_p-norm dot product

Recall that given any $p \geq 1$, the ℓ_p norm of a vector $x \in \mathbf{R}^n$ is $\ell_p(x) = \left(\sum_{i=1}^n |x(i)|^p\right)^{1/p}$ and that the cosine similarity of two non-negative vectors $x, y \in \mathbf{R}^n_+$ having unit ℓ_2 norm is $\sum_{i=1}^n x(i) \cdot y(i)$.

Furthermore, given $p \geq 1$, let

$$B_{p,n} := \left\{ x \in \mathbf{R}^n_+ \mid \sum_{i=1}^n x(i)^p \leq 1 \right\} \text{ and } S_{p,n} := \left\{ x \in \mathbf{R}^n_+ \mid \sum_{i=1}^n x(i)^p = 1 \right\},$$

be, respectively, the set of points contained in the p-ball of p-radius 1 with non-negative coordinates and the set of points lying on the p-sphere of p-radius 1 with non-negative coordinates.

The universe of the dot product similarity (that we define next) is $B_{p,n}$, which is uncountably infinite. To avoid technical issues in giving a minimally distorted LSH for this similarity, we restrict the universe $B_{p,n}$ to any finite subset $F_{p,n}$ of $B_{p,n}$. Given any such subset, the similarity $\text{DOT}_{p,n} : F_{p,n} \times F_{p,n} \to [0, \infty)$ is

$$\text{DOT}_{p,n}(x, y) = \sum_{i=1}^n x(i) \cdot y(i).$$

Notice that $\text{DOT}_{2,n}$ is the well-known cosine similarity (defined on the points of $S_{2,n}$). (Note that we have relaxed the notion of similarity to have range outside $[0, 1]$; the distortion bounds will take care of this issue. However, for the cosine similarity, the range is still $[0, 1]$.) We first show an upper bound on distortion and follow that with a matching lower bound.

▶ **Theorem 6.** *For $p \geq 2$,* $\text{distortion}(\text{DOT}_{p,n}) \leq 3n^{1-\frac{1}{p}}$.

Proof. The proof is omitted from this extended abstract. ◀

Now we show that the distortion of Theorem 6 is close to optimal using, once again, the center method.

▶ **Theorem 7.** *For $p \geq 1$,* $\mathsf{distortion}(\text{DOT}_{p,n}) \geq n^{1-\frac{1}{p}}$.

Proof. Consider the n vectors u_i defined as $u_i(i) = 1$, and $u_i(j) = 0$ for each $i \in [n]$ and for each $j \in [n] \setminus \{i\}$. Also, let $u_\star$ be the vector such that $u_\star(i) = n^{-\frac{1}{p}}$, for each $i \in [n]$, and let $X = \{u_1, u_2, \ldots, u_n\}$. Observe that for each $x \in X$, we have $\ell_p(x) = 1$ and $\ell_p(u_\star) = 1$.

Suppose that S is an LSHable similarity that distorts $\text{DOT}_{p,n}$ by the minimum possible amount. Since $S(u_i, u_j) = 0$ for every $i \neq j$, by Theorem 4 we know that there exists $u_i \in X$ such that $S(u_i, u_\star) \leq \frac{1}{n}$. Since $\text{DOT}_{p,n}(u_i, u_\star) = n^{-\frac{1}{p}}$, the distortion is at least $n^{1-\frac{1}{p}}$. ◀

As a simple corollary, we observe that the distortion for the cosine similarity is $\Theta(\sqrt{n})$ and that the distortion bound is tight for $p \geq 2$. We conjecture that it is generally tight for all $p \geq 1$, i.e., that Theorem 6 could be strengthened to all $p \geq 1$.

4.3 Sokal–Sneath similarities

Finally, we look at the Sokal–Sneath similarities. For $\gamma > 0$, let

$$\text{SOKAL-SNEATH}_\gamma(X, Y) = \frac{|X \cap Y| + \left|\overline{X \cup Y}\right|}{|X \cap Y| + \left|\overline{X \cup Y}\right| + \gamma\,|X \triangle Y|}.$$

Observe that SOKAL-SNEATH_1 is the Hamming similarity, $\text{SOKAL-SNEATH}_{1/2}$ is the Sokal–Sneath 1 similarity, and SOKAL-SNEATH_2 is the Rogers–Tanimoto similarity.

Rational set similarities [13] prove that $\text{SOKAL-SNEATH}_\gamma$ has an LSH iff $\gamma \geq 1$. Thus, the Hamming similarity and the Rogers–Tanimoto similarity admit an LSH, while the Sokal–Sneath 1 similarity does not admit an LSH.

We use the center method to prove a lower bound on the LSH-distortion of $\text{SOKAL-SNEATH}_\gamma$.

▶ **Theorem 8.** *For any $0 < \gamma < 1$,*

$$\frac{2}{1+\gamma} \leq \mathsf{distortion}(\text{SOKAL-SNEATH}_\gamma) \leq \frac{1}{\gamma}.$$

Proof. We begin with the lower bound. Given any ground set $[n]$ of even cardinality, consider the three sets $X = [n/2]$, $X' = [n] \setminus [n/2]$ and $Y = [n]$. We have, $\text{SOKAL-SNEATH}_\gamma(X, X') = 0$, $\text{SOKAL-SNEATH}_\gamma(X, Y) = \text{SOKAL-SNEATH}_\gamma(X', Y)$, and

$$\text{SOKAL-SNEATH}_\gamma(X, Y) = \frac{1/2}{1/2 + \gamma/2} = \frac{1}{1+\gamma}.$$

Consider any set similarity S on the ground set $[n]$ that admits an LSH, and that guarantees that $S(X, X') = 0$. By Theorem 4, there must exists $X^\star \in \{X, X'\}$ such that $S(X^\star, Y) \leq 1/2$. It follows that the distortion is at least $\frac{\frac{1}{1+\gamma}}{\frac{1}{2}} = \frac{2}{1+\gamma}$.

As for the upper bound, observe that for $0 < \gamma < 1$, we can approximate $\text{SOKAL-SNEATH}_\gamma$ with SOKAL-SNEATH_1 by introducing a distortion of $1/\gamma$. Since SOKAL-SNEATH_1 admits an LSH [13], it follows that $\mathsf{distortion}(\text{SOKAL-SNEATH}_\gamma) \leq 1/\gamma$. ◀

5 The k-sets Method

In this section we introduce our second tool for lower bounding the distortion of LSH. This method is geared towards set similarities. The main tool is the following theorem. Let $\mathcal{U}_{n,k}$ denote $\binom{[n]}{k}$.

▶ **Theorem 9.** *Let $k = o(\sqrt{n})$, and let $S : \mathcal{U}_{n,k} \times \mathcal{U}_{n,k} \to [0,1]$ be a similarity such that $S(X,Y) = 0$ if $X \cap Y = \varnothing$. If S admits an LSH, then*

$$f(S) := \operatorname*{avg}_{\substack{\{X,Y\} \in \binom{\mathcal{U}_{n,k}}{2} \\ |X \cap Y| = 1}} S(X,Y) \quad \leq \quad \alpha_k + O\left(\frac{k}{n}\right), \quad \text{where} \quad \alpha_k := \frac{1}{2k-1}.$$

This will be used in the following way. Suppose that we have a similarity S' defined on sets such that $S'(X,Y) = 0$ whenever X and Y are disjoint (not all, but many set similarities satisfy this property), and suppose also that $S'(X,Y) \geq d \cdot \alpha_k$ whenever X and Y are such that $|X| = |Y| = k$ and $|X \cap Y| = 1$. If S is LSHable, how small can its distortion be with respect to S'? By Theorem 9, there must exist a pair of sets such that $S(X,Y) \leq \alpha_k + O(k/n)$ which implies that the distortion of any LSHable S with respect to S' is at least $d - O(k^2/n)$.

In what follows, we begin with some technical Lemmas (§ 5.1) to prove Theorem 9 (§ 5.2) and then apply it, in § 5.3, to Braun–Blanquet similarity, establishing optimal distortion bounds for it.

5.1 Extremal partitions

A hash function h on $\mathcal{U}$ naturally induces a partition in the following sense: two objects $X, Y \in \mathcal{U}$ belong to the same side of the partition if $h(X) = h(Y)$. This view is particularly useful for our purposes and from now on we will identify a hash function with the partition that it induces.

▶ **Definition 10** (Acceptable partition)**.** A partition $\mathcal{P}$ of $\mathcal{U}_{n,k}$ *induces a pair* $\{X,Y\}$ (with $X \neq Y$) if X, Y belong to the same part of $\mathcal{P}$. A partition is *acceptable* if it induces no pair $\{X,Y\}$ such that X and Y are disjoint. The *value* of a partition is the number of pairs induced by it.

Our first goal is to prove that no acceptable partition of $\mathcal{U}_{n,k}$ has value greater than

$$\left(1 + O\left(k^2/n\right)\right) \cdot \frac{n^{2k-1}}{2(2k-1)((k-1)!)^2}.$$

▶ **Definition 11** (Nice partition)**.** An acceptable partition $\mathcal{P}$ of $\mathcal{U}_{n,k}$ is *nice* if it contains n parts $P_1, \ldots, P_t$, and if there exists a partition $I_1, \ldots, I_t$ of $[n]$ (with $I_i \neq \varnothing$ for $i \in [t]$, $\cup_{i=1}^{t} I_i = [n]$ and $I_i \cap I_j = \varnothing$ for each $\{i,j\} \in \binom{[t]}{2}$) such that, for each $i \in [t]$,

$$P_i = \left\{X \in \mathcal{U}_{n,k} \mid I_i \subseteq X \text{ and } X \cap \left(\cup_{j=1}^{i-1} I_j\right) = \varnothing\right\}.$$

We first show that nice partitions satisfy a slightly stronger version of the the above bound; we will then reduce any partition to a nice one.

▶ **Lemma 12.** *The value of a nice partition of $\mathcal{U}_{n,k}$ is at most*

$$\frac{n^{2k-1}}{2(2k-1)\left((k-1)!\right)^2}.$$

Proof. The proof is omitted from this extended abstract. ◀

We now make use of the following theorem of Hilton and Milner [23] (see [20] for a short proof), which bounds the maximum cardinality of an Erdös–Ko–Rado [19] family that is not a star.

▶ **Theorem 13** (Hilton–Milner [23]). *Let $\mathcal{F} \subseteq \mathcal{U}_{n,k}$ be a family of sets with pairwise non-empty intersection with $n \geq 2k$. If $\bigcap_{F\in\mathcal{F}} F = \varnothing$ then $|\mathcal{F}| \leq \binom{n-1}{k-1} - \binom{n-k-1}{k-1} + 1$.*

▶ **Fact 14.** $\binom{n-1}{k-1} - \binom{n-k-1}{k-1} + 1 \leq O\left(k \cdot \frac{n^{k-2}}{(k-2)!}\right)$.

▶ **Lemma 15.** *The value of an acceptable partition of $\mathcal{U}_{n,k}$ is at most*

$$\left(1 + O\left(\frac{k^2}{n}\right)\right) \cdot \frac{n^{2k-1}}{2(2k-1)\left((k-1)!\right)^2}.$$

Proof. The proof is omitted from this extended abstract. ◀

5.2 Proof of Theorem 9

Proof. Let $\alpha = \operatorname{avg}_{\substack{\{X,Y\}\in\binom{\mathcal{U}_{n,k}}{2} \\ |X\cap Y|=1}} S(X,Y)$ be the average similarity between pairs of sets of cardinality k having an intersection of cardinality 1. Let σ be the total amount of similarity between unordered pairs of sets of cardinality k having intersection 1. It is equal to:

$$\sigma = n\frac{\binom{n-1}{k-1}\binom{n-k}{k-1}}{2}\alpha.$$

Recall that, in general, we have that

$$\binom{n}{\ell} \geq \frac{(n-\ell)^\ell}{\ell!} = \frac{n^\ell\left(1-\frac{\ell}{n}\right)^\ell}{\ell!} \geq \frac{n^\ell}{\ell!}\left(1-\frac{\ell^2}{n}\right).$$

Substituting k for ℓ, we obtain:

$$\sigma \geq \left(1 - O\left(\frac{k^2}{n}\right)\right)\frac{n^{2k-1}}{2((k-1)!)^2}\cdot\alpha,$$

where the $O(\cdot)$ term tends to 0, since $k = o(\sqrt{n})$. Since $S(X,Y) = 0$ whenever $|X\cap Y| = 0$, we cannot give positive probability to a hash function placing two such sets X and Y in the same part, for otherwise we would have infinite distortion. Hence, we can only use acceptable partitions. Suppose that the S has an LSH and assume wlog that this LSH gives positive probabilities $p_1,\ldots,p_h > 0$ to partitions $P_1,\ldots,P_h$, and that it gives probability 0 to other partitions. Let $v_1,\ldots,v_h$ be the values of partitions $P_1,\ldots,P_h$, and observe that $\sum_{i=1}^h p_i = 1$. Then, we have

$$\sigma = \sum_{\substack{\{X,Y\}\in\binom{\mathcal{U}_{n,k}}{2} \\ |X\cap Y|=1}} S(X,Y) = \sum_{i=1}^{h}(p_i v_i),$$

i.e., the total amount of similarity mass that an acceptable partition brings to our similarity's values is equal to the probability that the LSH assigns to the partition times the number of the partition's pairs, equivalently, its own value. By Lemma 15, the value of an acceptable partition is at most

$$\tau = \left(1 + O\left(\frac{k^2}{n}\right)\right)\frac{n^{2k-1}}{2(2k-1)\left((k-1)!\right)^2}.$$

Therefore, $\sigma \leq \sum_{i=1}^{h} (\tau p_h) = \tau$. I.e., if S admits an LSH, then $\tau \geq \sigma$. Thus, we must have

$$1 \geq \frac{\sigma}{\tau} \geq \left(1 - O\left(\frac{k^2}{n}\right)\right) \frac{\frac{n^{2k-1}}{2((k-1)!)^2} \cdot \alpha}{\frac{n^{2k-1}}{2(2k-1)((k-1)!)^2}} = \left(1 - O\left(\frac{k^2}{n}\right)\right) \alpha \cdot (2k-1),$$

which implies

$$\alpha \leq \left(1 + O\left(\frac{k^2}{n}\right)\right) \frac{1}{2k-1} = \frac{1}{2k-1} + O\left(\frac{k}{n}\right). \quad \blacktriangleleft$$

5.3 The distortion of Braun–Blanquet

Recall the definition of Braun–Blanquet, that operates on the subsets of the ground set $[n]$:

$$\text{BRAUN-BLANQUET}(X, Y) = \frac{|X \cap Y|}{\max(|X|, |Y|)},$$

if $|X| + |Y| \geq 1$, and $\text{BRAUN-BLANQUET}(X, Y) = 1$ if $X = Y = \varnothing$.

Observe that, for sets $X, Y \subseteq [n]$ such that $|X| = |Y| = k \geq 1$, both Braun–Blanquet and Sørensen–Dice evaluate to $1/k$ if $|X \cap Y| = 1$, and that they evaluate to 0 when $|X \cap Y| = 0$. Therefore, Theorem 9 implies that they have to be distorted by at least $(1 - o_n(1)) \cdot (2 - 1/k)$ when applied on such pairs of k-sets. By letting k grow to infinity, we obtain an asymptotically tight lower bound of 2 on their distortions. More precisely, by selecting $k = \Theta\left(n^{1/3}\right)$, and by letting n grow to infinity, their distortion is at least $2 - \Theta\left(n^{-1/3}\right)$. If we denote with S any of the two similarities, with S' any LSHable similarity with the same domain, and with X, Y any two sets that minimize $S'(X, Y)$, we obtain,

$$\frac{S(X,Y)}{S'(X,Y)} \geq \frac{\frac{1}{k}}{\frac{1}{2k-1} + O(\frac{k}{n})} = \frac{2 - \frac{1}{k}}{1 + O(\frac{k^2}{n})} = 2 - O(n^{-1/3}).$$

We finally observe that min-wise independent permutations [8, 9] achieve a distortion of $2 - \Theta\left(n^{-1}\right)$ for Braun–Blanquet. Thus, we have the following theorem:

▶ **Theorem 16.** $\text{distortion}(\text{BRAUN-BLANQUET}) = 2 - o(1)$.

▶ **Theorem 17.** $1 - \text{BRAUN-BLANQUET}$ *can be isometrically embedded into* ℓ_1.

Proof. The proof is omitted from this extended abstract. ◀

Thus, BRAUN-BLANQUET passes both (T1) and (T2) (see the introduction), and yet it is not LSH-able.

6 Ad hoc Approaches

In this section we discuss another similarity, whose distortion bound we prove through a simple ad hoc approach.

6.1 Forbes similarity

The Forbes similarity is defined as $\text{FORBES}(X, Y) = n \cdot \frac{|X \cap Y|}{|X|\,|Y|}$ if $|X|, |Y| \geq 1$, $\text{FORBES}(X, \varnothing) = 0$ if $|X| \geq 1$, and if $\text{FORBES}(\varnothing, \varnothing) = 1$. Since $F(\{1\}, \{1\}) = n$, we have the following simple observation.

▶ **Theorem 18.** distortion(FORBES) $= n$.

Proof. The lower bound is trivial since FORBES$(\{1\},\{1\}) = n$ and no LSH can assign a value larger than 1 to a pair of sets.

We give an LSH for the similarity FORBES$/n$, thus proving an upper bound of n on its distortion. The hash function h will be chosen as follows: $h(\varnothing) = \varnothing$ and, for each $X \neq \varnothing$ independently, $h(X)$ will be chosen uniformly at random from the elements of X. Then, if $X \neq Y$, we have $\Pr[h(X) = h(Y)] = \frac{|X \cap Y|}{|X| \cdot |Y|}$. ◀

7 Experiments

In this section we report on the outcome of two types of experiments. As we have seen in the previous sections the distortion of Braun–Blanquet and of Sørensen–Dice is $2 - o(1)$ and this bound can be matched by Jaccard, which is LSHable. Distortion being a worst-case notion, it is conceivable that the typical behavior of Jaccard with real-world datasets could be somewhat better. This is exactly what our experiments with three real world data sets show. We stress that our results are preliminary, but they give reasons for hope and might justify a more comprehensive experimental assessment. The average distortion turns out to be as low as 1.3 for some of our data sets and always less than two. The second set of experiments is a feasibility study of the LSH scheme for Anderberg and Rogers–Tanimoto, similarities that until recently were not known to be LSHable. As shown in [13] they are, but in a somewhat peculiar way, for the LSH schemes might need exponentially many bits (with low probability). The goal of our tests is to see whether such schemes are practical. Our study shows that they are and that in fact they can be very effective with very few bits. We begin by describing our data sets.

7.1 Datasets

We use three publicly available datasets: (i) a collection of more than 110K scientific papers downloaded from CiteSeerX, (ii) 29K scientific articles downloaded from ArXiv, and (iii) 104K Wikipedia articles. The collection of XML metadata of CiteSeerX and ArXiv where accessed using the OAI protocol for metadata harvesting, which is supported by both digital libraries. The Wikipedia collection was obtained from `en.wikipedia.org/wiki/Wikipedia:Database_download`. The words in each paper were transformed into lowercase and each document became a bag of words (no repetitions).

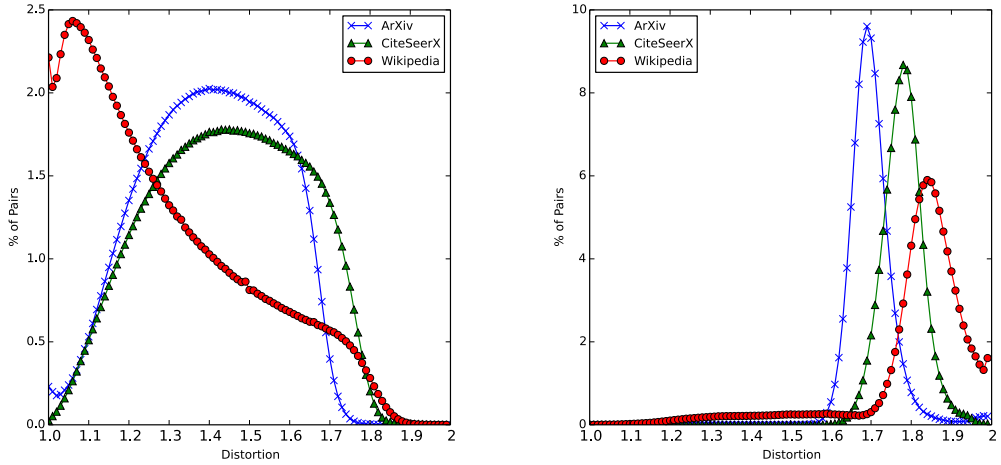
For the experiments of § 7.3 the documents underwent the following "cleaning" procedure: (i) all words not included in top 1000 most frequent words of the whole dataset were removed and, (ii) every word was hashed to a unique integer. As a result, the papers are represented as vectors containing integers in the range $[1000] = \{1, 2, \ldots, 1000\}$.

7.2 Distortion on real data

From each corpus, we selected 50 million random pairs of documents and computed the distortion, i.e., the ratio between the Jaccard value (computed exactly) and the two similarities Braun–Blanquet and Sørensen–Dice. Figure 2a shows the distortion w.r.t. Braun–Blanquet for our three datasets ArXiv, CiteSeer, and Wikipedia. For each value of the distortion on the x-axis, the plot gives, on the y-axis, the fraction of pairs with that distortion. Similarly, Figure 2b shows the distortion w.r.t. Sørensen–Dice. Table 2 displays the average distortion and the variance of these experiments.

Table 2 Experimental results.

	Braun–Blanquet		Sørensen–Dice	
	μ	σ	μ	σ
ArXiv	1.45	0.2	1.78	0.09
CiteSeerX	1.4	0.16	1.7	0.05
Wikipedia	1.29	0.21	1.81	1.14



(a) Braun–Blanquet similarity. **(b)** Sørensen–Dice similarity.

Figure 2 Percentage of document pairs with distortion δ with respect to shingles as δ increases.

Overall, these tests show that in real-world scenarios the average distortion of Braun–Blanquet and Sørensen–Dice can be significantly smaller than the worst case bound.

7.3 LSH schemes for rational set similarities

Let us start by recalling the definitions of the similarities we deal with in this section. The Anderberg similarity is defined as follows. Given two nonempty sets X, Y of n elements,

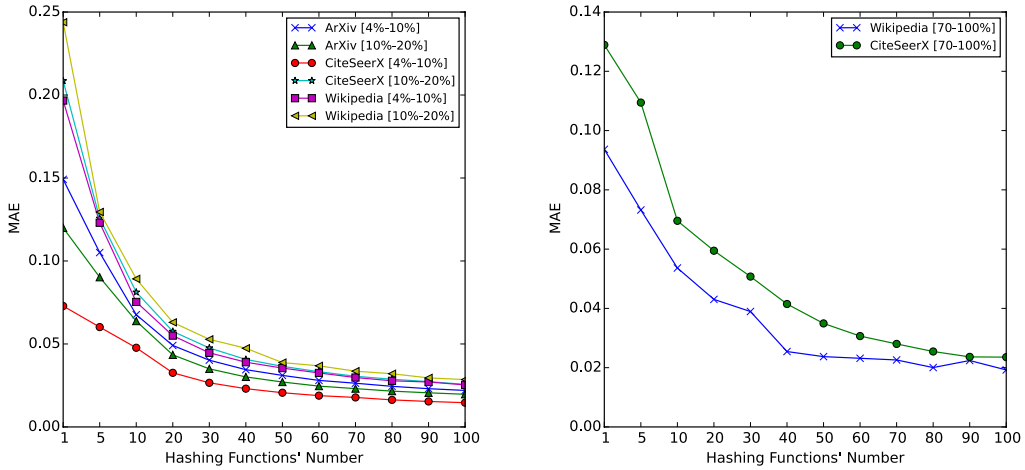
$$\text{ANDERBERG}(X,Y) = \frac{|X \cap Y|}{|X \cap Y| + 2|X \triangle Y|},$$

where $\triangle$ is the symmetric difference. (Note that S_2 is the Anderberg similarity.) The value is zero if exactly one of the two sets is empty, and it is 1 whenever $X = Y$. In [13] it is proven that the following is an LSH scheme for it. Pick a positive integer r at random with probability 2^{-r}. Let $h_1, \ldots, h_r$ be r shingles picked independently. Then, $h(X) := (h_1(X), \ldots, h_r(X))$ is an LSH scheme for A, i.e., $\text{ANDERBERG}(X,Y) = \Pr[h(X) = h(Y)]$.

The Rogers–Tanimoto similarity is defined as

$$\text{ROGERS-TANIMOTO}(X,Y) = \frac{|X \cap Y| + \left|\overline{X \cup Y}\right|}{|X \cap Y| + \left|\overline{X \cup Y}\right| + 2|X \triangle Y|}.$$

(Note that H_2 is the Rogers–Tanimoto similarity.) The following is the LSH scheme for Rogers–Tanimoto proposed in [13]. Pick r as before, and then pick r elements $e_1, e_2, \ldots, e_r$



(a) Anderberg similarity. **(b)** Rogers-Tanimoto similarity.

Figure 3 Mean Average Error as number of hash functions applied varies.

from the ground set independently at random. The random hash function h is defined as follows. For a set X, we let $h(X) := (e_1 \in X, \ldots, e_r \in X)$ (where $e_i \in X$ is a boolean value). Given two sets X and Y, $h(X) = h(Y)$ iff the two vectors coincide on each coordinate (for each element $e = e_1, e_2, \ldots, e_r$, either both sets have it or they both do not).

Recall that in this experiment our corpora consists of bag of words in which only the one thousand most popular words are retained. So each document can be thought of as binary vector of one thousand coordinates (coordinate i is one iff the ith most popular word is in the document).

The experiment is as follows. Let h denote a generic hash function of the LSH scheme that we are testing. From each corpus, we picked one hundred thousand random pairs of documents. Then, for every $k \in [100]$, we selected k hash functions $h_1, \ldots, h_k$ and estimated the similarity of the random pair in the usual fashion, i.e., as the fraction of times that $h_i(X) = h_i(Y)$, for $i \in [k]$.

Figure 3a shows, for each value of k on the x-axis, the mean absolute error (MAE) w.r.t. the real value of Anderberg. Note that already for $k = 20$ the MAE is below 0.05. Since the expected number of shingles used in each h is two (with very small variance) this shows the LSH scheme is inexpensive both time-wise and space-wise. Similar conclusions apply to Rogers–Tanimoto, as Figure 3b shows.

The experimental results show that the MAE decreases as the number of hashing functions applied increase for each of the databases and similarities tested, reinforcing the theoretical aspects of LSH applied to specific group of similarities that admit such an LSH.

8 Conclusions

In this paper we studied the notion of distorted locality sensitive hashing schemes for a number of widely-used similarities that do not admit exact such schemes. For most of them, we have obtained tight bounds on the minimum distortion required for obtaining an LSH. In doing so, we developed two lower bounding tools that could be useful for bounding the distortion of other similarities that are not LSHable.

To complement our theoretical bounds, we also studied the behavior of our proposed distorted LSH schemes on real datasets. Our main observation is that in practice, the average distortion is milder than what is dictated by the worst-case bounds.

It will be interesting to consider other non-LSHable similarities and study their distortion. The encyclopedia [15] is a rich source for such similarities.

Acknowledgments. We thank the anonymous reviewers for several useful comments and suggestions.

References

1 Michael R. Anderberg. *Cluster Analysis for Applications*. Academic Press, Inc., New York, 1973.

2 Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *FOCS*, pages 459–468, 2006.

3 Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *C. ACM*, 51(1):117–122, 2008.

4 Ismail Avcibaş, Mehdi Kharrazi, Nasir Memon, and Bülent Sankur. Image steganalysis with binary similarity measures. *EURASIP Journal on Applied Signal Processing*, 2005:2749–2757, 2005.

5 Lalit R. Bahl, John Cocke, Frederick Jelinek, and Josef Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE TOIT*, 20(2):284–287, 1974.

6 Charles H. Bennett and Peter W. Shor. Quantum information theory. *IEEE TOIT*, 44(6):2724–2742, 1998.

7 Josias Braun. *Die Vegetationsverhältnisse der Schneestufe in den Rätisch-Lepontischen Alpen: Ein Bild des Pflanzenlebens an seinen äussersten Grenzen.* Schweizerische Naturforschende Gesellschaft, 1913.

8 Andrei Z. Broder. On the resemblance and containment of documents. In *Proc. SEQUENCES*, pages 21–29, 1997.

9 Andrei Z. Broder, Moses Charikar, Alan M. Frieze, and Michael Mitzenmacher. Min-wise independent permutations. *JCSS*, 60(3):630–659, 2000.

10 Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

11 Julie Chabalier, Jean Mosser, and Anita Burgun. A transversal approach to predict gene product networks from ontology-based similarity. *BMC Bioinformatics*, 8(1):235, 2007.

12 Moses Charikar. Similarity estimation techniques from rounding algorithms. In *Proc. STOC*, pages 380–388, 2002.

13 Flavio Chierichetti and Ravi Kumar. LSH-preserving functions and their applications. *J. ACM*, 62(5):33:1–33:25, 2015.

14 Flavio Chierichetti, Ravi Kumar, and Mohammad Mahdian. The complexity of LSH feasibility. *Theoretical Computer Science*, 530:89–101, 2014.

15 MichelMarie Deza and Elena Deza. *Encyclopedia of Distances*. Springer Berlin Heidelberg, 2009. doi:10.1007/978-3-642-00234-2_1.

16 Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

17 Sarah J Dixon, Nina Heinrich, Maria Holmboe, Michele L Schaefer, Randall R Reed, Jose Trevejo, and Richard G Brereton. Use of cluster separation indices and the influence of outliers: application of two new separation indices, the modified silhouette index and the overlap coefficient to simulated data and mouse urine metabolomic profiles. *Journal of Chemometrics*, 23(1):19–31, 2009.

18 William HR Equitz and Thomas M Cover. Successive refinement of information. *IEEE TOIT*, 37(2):269–275, 1991.

19 Paul Erdös, Chao Ko, and Richard Rado. Intersection theorems for systems of finite sets. *Quart. J. Math. Oxford*, 12(2):313–320, 1961.

20 Péter Frankl and Zoltan Füredi. Non-trivial intersecting families. *JCT, Series A*, 41(1):150–153, 1986. URL: `doi:10.1016/0097-3165(86)90121-4`.

21 George W Furnas, Scott Deerwester, Susan T Dumais, Thomas K Landauer, Richard A Harshman, Lynn A Streeter, and Karen E Lochbaum. Information retrieval using a singular value decomposition model of latent semantic structure. In *SIGIR*, pages 465–480, 1988.

22 Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, pages 518–529, 1999.

23 A.J.W. Hilton and E.C. Milner. Some intersection theorems for systems of finite sets. *Quart. J. Math. Oxford*, 18:369–384, 1967.

24 Piotr Indyk and Jiri Matousek. Low-distortion embeddings of finite metric spaces. *Handbook of Discrete and Computational Geometry*, pages 177–196, 2004.

25 Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC*, pages 604–613, 1998.

26 Piotr Indyk, Rajeev Motwani, Prabhakar Raghavan, and Santosh Vempala. Locality-preserving hashing in multidimensional spaces. In *STOC*, pages 618–625, 1997. `doi:10.1145/258533.258656`.

27 Dennis H Knight. A phytosociological analysis of species-rich tropical forest on Barro Colorado Island, Panama. *Ecological Monographs*, pages 259–284, 1975.

28 Patricia Koleff, Kevin J Gaston, and Jack J Lennon. Measuring beta diversity for presence–absence data. *Journal of Animal Ecology*, 72(3):367–382, 2003.

29 J Looman and JB Campbell. Adaptation of Sorensen's k (1948) for estimating unit affinities in prairie vegetation. *Ecology*, pages 409–416, 1960.

30 Qin Lv, William Josephson, Zhe Wang, Moses Charikar, and Kai Li. Multi-probe LSH: efficient indexing for high-dimensional similarity search. In *VLDB*, pages 950–961, 2007.

31 EMM Manders, FJ Verbeek, and JA Aten. Measurement of co-localization of objects in dual-colour confocal images. *Journal of Microscopy*, 169(3):375–382, 1993.

32 Rada Mihalcea, Courtney Corley, and Carlo Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*, pages 775–780, 2006.

33 MED Poore. The use of phytosociological methods in ecological investigations: I. The Braun–Blanquet system. *The Journal of Ecology*, pages 226–244, 1955.

34 David J Rogers and Taffee T Tanimoto. A computer program for classifying plants. *Science*, 132(3434):1115–1118, 1960.

35 Pavel Rychlỳ. A lexicographer-friendly association score. In *RASLAN*, pages 6–9, 2008.

36 Gerard Salton. Developments in automatic text retrieval. *Science*, 253(5023):974, 1991.

37 Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.

38 Venu Satuluri and Srinivasan Parthasarathy. Bayesian locality sensitive hashing for fast similarity search. *VLDB*, 5(5):430–441, 2012.

39 AVI Shmida and Mark V Wilson. Biological determinants of species diversity. *Journal of Biogeography*, pages 1–20, 1985.

40 Peter H. A. Sneath and Robert R Sokal. *Numerical Taxonomy: The Principles and Practice of Numerical Classification.* W. H. Freeman, 1973.

41 PHA Sneath and R Johnson. The influence on numerical taxonomic similarities of errors in microbiological tests. *Journal of General Microbiology*, 72(2):377–392, 1972.

42 Thorvald Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biol. Skr.*, 5:1–34, 1948.

43 Jingdong Wang, Heng Tao Shen, Jingkuan Song, and Jianqiu Ji. Hashing for similarity search: A survey. Technical report, CoRR abs/1408.2927, 2014.

44 Robert H Whittaker. Evolution and measurement of species diversity. *Taxon*, pages 213–251, 1972.

45 ST Williams, M Goodfellow, G Alderson, EMH Wellington, PHA Sneath, and MJ Sackin. Numerical classification of Streptomyces and related genera. *Journal of General Microbiology*, 129(6):1743–1813, 1983.

46 SK Michael Wong, Wojciech Ziarko, and Patrick CN Wong. Generalized vector spaces model in information retrieval. In *SIGIR*, pages 18–25, 1985.

Constructive Non-Commutative Rank Computation Is in Deterministic Polynomial Time*

Gábor Ivanyos[1], Youming Qiao[2], and K Venkata Subrahmanyam[3]

1 **Institute for Computer Science and Control, Hungarian Academy of Sciences, Budapest, Hungary**
Gabor.Ivanyos@sztaki.mta.hu

2 **Centre for Quantum Software and Information,University of Technology Sydney, Australia**
Youming.Qiao@uts.edu.au

3 **Chennai Mathematical Institute, Chennai, India**
kv@cmi.ac.in

Abstract

Let $\mathcal{B}$ be a linear space of matrices over a field $\mathbb{F}$ spanned by $n \times n$ matrices $B_1, \ldots, B_m$. The non-commutative rank of $\mathcal{B}$ is the minimum $r \in \mathbb{N}$ such that there exists $U \leq \mathbb{F}^n$ satisfying $\dim(U) - \dim(\mathcal{B}(U)) \geq n - r$, where $\mathcal{B}(U) := span(\cup_{i\in[m]} B_i(U))$.

Computing the non-commutative rank generalizes some well-known problems including the bipartite graph maximum matching problem and the linear matroid intersection problem.

In this paper we give a deterministic polynomial-time algorithm to compute the non-commutative rank over any field $\mathbb{F}$. Prior to our work, such an algorithm was only known over the rational number field $\mathbb{Q}$, a result due to Garg et al, [20]. Our algorithm is constructive and produces a witness certifying the non-commutative rank, a feature that is missing in the algorithm from [20].

Our result is built on techniques which we developed in a previous paper [24], with a new reduction procedure that helps to keep the blow-up parameter small. There are two ways to realize this reduction. The first involves constructivizing a key result of Derksen and Makam [12] which they developed in order to prove that the null cone of matrix semi-invariants is cut out by generators whose degree is polynomial in the size of the matrices involved. We also give a second, simpler method to achieve this. This gives another proof of the polynomial upper bound on the degree of the generators cutting out the null cone of matrix semi-invariants.

Both the invariant-theoretic result and the algorithmic result rely crucially on the regularity lemma proved in [24]. In this paper we improve on the constructive version of the regularity lemma from [24] by removing a technical coprime condition that was assumed there.

1998 ACM Subject Classification F.2.0 General, F.2.1 Numerical Algorithms and Problems, I.1.2 Algorithms , F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases invariant theory, non-commutative rank, null cone, symbolic determinant identity testing, semi-invariants of quivers

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.55

* Research of the first author was also supported in part by the Hungarian National Research, Development and Innovation Office – NKFIH Grant K115288. Youming's research was supported by the Australian Research Council DECRA DE150100720. KV's research was supported by a grant from the Infosys foundation.

© Gábor Ivanyos, Youming Qiao, and Venkata Subrahmanyam;
licensed under Creative Commons License CC-BY

1 Introduction

1.1 From the bipartite perfect matching problem to the commutative and non-commutative rank problems

Given a bipartite graph $G = (L \cup R, E)$ where $|L| = |R|$, the celebrated Hall's marriage theorem states that G has a perfect matching if and only if G has no *shrunk subsets*: $S \subseteq L$ is called a shrunk subset, if $|N(S)| < |S|$ where $N(S)$ denotes the set of neighbours of S.

Consider the following linear algebraic analogue of the bipartite perfect matching problem. Let $\mathbb{F}$ be a field, and let $U \cong V \cong \mathbb{F}^n$ be two vector spaces. We assume $\mathbb{F}$ is large (larger than a fixed polynomial in n is enough). U and V may be thought of as corresponding to sets of vertices. In a bipartite graph, each edge may be viewed as a "partial function" from the left vertex set to the right vertex set. Carrying out this analogue, in the linear algebraic setting we shall think of one linear map from U to V as one edge. After fixing bases of U and V this linear map is represented as a matrix. Let $M(n, \mathbb{F})$ denote the linear space of $n \times n$ matrices over $\mathbb{F}$. Then the edge set in the linear algebraic setting is just a set of matrices, and suppose we have m matrices $\{B_1, \ldots, B_m\} \subseteq M(n, \mathbb{F})$.

In a bipartite graph, a perfect matching is a subset of edges that form a bijective function from the left vertex set to the right vertex set. Choosing a subset of edges can be viewed as assigning 0 and 1 to the edges and then selecting those edges with 1's. This leads us to consider linear combinations of the matrices in the linear algebraic setting. That is, we are interested in the linear span of the given matrices $B_1, \ldots, B_m \in M(n, \mathbb{F})$, denoted as $\mathcal{B} = \text{span}(B_1, \ldots, B_m) \leq M(n, \mathbb{F})$. We call a linear subspace of $M(n, \mathbb{F})$ a *matrix space*. A "perfect matching" in the linear algebraic setting is then naturally a full-rank (non-singular) matrix as it is a bijective linear map between the left and the right vector spaces. A matrix space is *non-singular* if it contains a non-singular matrix, and *singular* otherwise.

It is also easy to obtain an analogue of shrunk subsets as in the Hall's marriage theorem. For $c \in \mathbb{N}$, we call $W \leq U$ a *c-shrunk subspace* of $\mathcal{B} = \text{span}(B_1, \ldots, B_m) \leq M(n, \mathbb{F})$, if $\dim(W) - \dim(\mathcal{B}(W)) \geq c$ where $\mathcal{B}(W) := \text{span}(\cup_{i \in [m]} B_i(W))$. We also call $W \leq U$ a shrunk subspace if it is a c-shrunk subspace for some $c \geq 1$. Clearly, if $\mathcal{B}$ possesses a shrunk subspace then it is singular.

A natural question is then whether the analogue of Hall's theorem holds, that is, whether every singular matrix space has a shrunk subspace. A counter-example is not hard to find, as evidenced by the linear space of 3×3 skew-symmetric matrices. That is, while perfect matchings and shrunk subsets are two sides of the same coin for bipartite graphs, non-singular matrices and shrunk subspaces are not in the linear algebraic setting. This gives rise to two natural algorithmic problems.

Input Given $B_1, \ldots, B_m \in M(n, \mathbb{F})$, let $\mathcal{B}$ be the matrix space spanned by these matrices.

The commutative rank problem Call the maximum rank over matrices in $\mathcal{B}$ the *commutative rank* of $\mathcal{B}$, denoted as $\text{crk}(\mathcal{B})$. The commutative rank problem asks to compute $\text{crk}(\mathcal{B})$.

The non-commutative rank problem Define the non-commutative corank to be the maximum $c \in \mathbb{N}$ such that there exists a c-shrunk subspace of $\mathcal{B}$, and the *non-commutative rank* to be $n - c$, denoted as $\text{ncrk}(\mathcal{B})$. The non-commutative rank problem asks to compute $\text{ncrk}(\mathcal{B})$.

The names of these problems were coined in Fortin and Reutenauer [19],[1] though both problems have been around since 1970's (see Section 1.2).

While these two problems are different in general, as suggested by the 3×3 skew-symmetric matrix space, they do coincide in some special cases. For this let us recall how the bipartite perfect matching problem and the linear matroid intersection problem can be cast as special instances of both problems.

Given a bipartite graph $G = (L \cup R, E)$ where $L = R = \{1, 2, \ldots, n\}$, for each edge $e = (i, j) \in E$ construct the elementary matrix $E_{i,j}$, which span a matrix space $\mathcal{B}$.

It is easy to see that the size of a maximum matching equals the maximum rank over matrices in $\mathcal{B}$ (over a large enough field). Hall's marriage theorem then implies that if the maximum rank is $n - c$ then there exists a c-shrunk subspace, and it is clear that if there exists a c-shrunk subspace then the maximum rank can be no larger than $n - c$. This shows that the commutative rank and the non-commutative rank coincide in this special case.

Lovász [27] observed that the linear matroid intersection problem (LMIP) can be cast as an instance of both the commutative and the non-commutative rank problems as follows. The input to LMIP is two tuples of vectors $(a_1, \ldots, a_m)$ and $(b_1, \ldots, b_m)$ where $a_i, b_j \in \mathbb{F}^n$. It asks to compute the maximum $s \in \mathbb{N}$ such that there exist $1 \le i_1 < \cdots < i_s \le m$ with both $\{a_{i_1}, \ldots, a_{i_s}\}$ and $\{b_{i_1}, \ldots, b_{i_s}\}$ being linearly independent. Construct m rank-1 matrices $a_i b_i^t$ spanning a matrix space $\mathcal{B}$. Using Edmonds' matroid intersection theorem, Lovász showed that in this case the commutative rank and the non-commutative rank of $\mathcal{B}$ coincide, and both are equal to the matroid intersection number.

1.2 Backgrounds to the commutative and non-commutative rank problems

As far as we are aware the commutative rank problem was first proposed by Edmonds [17] in 1967. It was then realized that it admits an efficient randomized algorithm via the Schwartz-Zippel lemma. Lovász [27] cast several problems from matroid theory, including the linear matroid intersection problem and the linear matroid parity problem, as special instances of the commutative rank problem. To decide whether the commutative rank is full is now better known as the *symbolic determinant identity testing* (SDIT) problem.

At present, SDIT stands at the frontier of proving arithmetic circuit lower bounds due to the celebrated work of Kabanets and Impagliazzo [25], which was recently improved by Carmosino et al [5]. They show that a deterministic efficient algorithm for SDIT implies the existence of a polynomial family such that its graph is in NE, but it cannot be computed by polynomial-size arithmetic circuits. Also, since determinants with affine entries are equivalent[33] to weakly-skew arithmetic circuits,[2], up to a polynomial overhead, SDIT is just another way of formulating the more well-known *polynomial identity testing* (PIT) problem for weakly-skew arithmetic circuits. The research into PIT has received a lot of attention since early 2000 (see the surveys [32, 30]).

[1] We explain the name choices here: from $B_1, \ldots, B_m \in M(n, \mathbb{F})$ one can construct a symbolic matrix $T = x_1B_1 + \cdots + x_mB_m$ where each entry is a linear form. When the variables x_i's commute, the rank of T over the rational function field is equal to the commutative rank (as we assumed $\mathbb{F}$ is large enough). If x_i's are non-commutative, there exists a non-commutative analogue of the rational function field, called the free skew field. Fortin and Reutenauer [19] (building on [8]) proved that the rank of T over the free skew field is equal to the non-commutative rank defined above.

[2] An arithmetic circuit is weakly skew if each product gate is of fan-in 2 and has at least one child labelled by a variable or a field element. The computation power of weakly skew circuit is between arithmetic formulas and arithmetic circuits.

The non-commutative rank problem can be traced back to 1970's, when Cohn[6, 7] raised this problem in his research on the free skew field.[3] There Cohn showed that this problem is decidable. This is already non-trivial, partly because his starting point was the free skew field, which itself is a complex object. This was improved to PSPACE by Cohn and Reutenauer [9] by reducing to testing the solvability of a system of multivariate polynomial equations. Fortin and Reutenauer [19] realized the connection between the non-commutative rank problem and the structure of matrix spaces,[4] and the definition of the non-commutative rank in terms of shrunk subspaces we have given above is due to them. Around the same time, Gurvits [21] came to the non-commutative rank problem in his study of the commutative rank problem, and endowed both the commutative and non-commutative rank problems with quantum information theoretic interpretations. In 2014, Hrubeš and Wigderson [22] studied non-commutative arithmetic formulas with divisions, and reduced the identity testing problem in this model to the non-commutative rank problem. In particular, they discovered that a good upper bound of a quantity in invariant theory (which will be explained below) implies the efficient elimination of divisions in arithmetic formulas with divisions, as well as an efficient randomized algorithm for the non-commutative rank problem.

We refer readers to [24, 20] for an introduction to the various avatars of the non-commutative rank problem.

Our main motivation to examine the non-commutative rank problem comes from its relation with the commutative one. For the purpose of getting arithmetic circuit lower bounds following [5], it is actually enough to put SDIT in NP, that is, to find a short witness which testifies the singularity of singular matrix spaces. Shrunk subspaces just form one natural class of singularity witnesses, as evidenced in its analogue with the shrunk subsets in Hall's marriage theorem and its crucial role in the linear matroid intersection problem. If we can efficiently distinguish matrix spaces with shrunk subspaces from those without then, for SDIT it would be enough to focus on singular matrix spaces without shrunk subspaces. While such spaces are known to have a rich structure, see [2, 18, 16], to further understand them and see how they can be efficiently recognized, is a natural approach to study SDIT.

1.3 Certifying matrix spaces without shrunk subspaces

Since 2015 there has been impressive progress on the non-commutative rank problem and the relevant invariant theoretic problem. To introduce this progress, let us review the link between the non-commutative rank problem and invariant theory.

Recall that the non-commutative rank problem asks to compute, given a matrix space $\mathcal{B} \leq M(n, \mathbb{F})$, the maximum c such that $\mathcal{B}$ has a c-shrunk subspace. For convenience, let us consider a decision version of this problem, that is to decide whether $\mathcal{B}$ has a shrunk subspace. One conceptual difficulty in tackling this problem is to find a short witness certifying that a matrix space $\mathcal{B}$ has no shrunk subspaces. It is clear that, if $\mathcal{B}$ has a non-singular matrix, then this non-singular matrix can be used to certify that $\mathcal{B}$ has no shrunk subspaces. But we've seen that, the space of 3×3 skew-symmetric matrices sk_3, has no non-singular matrices, nor shrunk subspaces. Our approach to resolving this difficulty is to introduce the following blow-up operation for matrix spaces. Given a matrix space $\mathcal{B} \leq M(n, \mathbb{F})$, the dth *blow-up* of

[3] The free skew field may be thought of as the non-commutative analogue of the rational function field.

[4] Fortin and Reutenauer were inspired by a paper by Eisenbud and Harris [18] in algebraic geometry. In fact matrix spaces with commutative rank bounded from above roughly correspond to certain torsion-free sheaves on projective spaces.

$\mathcal{B}$, denoted as $\mathcal{B}^{[d]}$, is defined to be the matrix space $M(d,\mathbb{F})\otimes\mathcal{B}\leq M(nd,\mathbb{F})$. It is easy to show the following.

▶ **Proposition 1.** *If $\mathcal{B}\leq M(n,\mathbb{F})$ has a c-shrunk subspace, then $\mathcal{B}^{[d]}$ has a cd-shrunk subspace.*

That is, the blow-up operation preserves shrunk subspaces. On the other hand, it is not immediately clear what the blow-up operation's effect is on matrix spaces without shrunk subspaces, though it can be shown that for sk_k, the space of skew-symmetric matrices of an odd size k, $\mathrm{sk}_k{}^{[2]}$ contains a non-singular matrix. Therefore the first question is whether for any matrix space $\mathcal{B}\leq M(n,\mathbb{F})$ without shrunk subspaces, $\mathcal{B}^{[d]}$ contains a non-singular matrix for some finite d. This turns out to be true, but to see it we need to go through several results in invariant theory and algebraic geometry.

Consider the group action of $(A,C)\in \mathrm{SL}(n,\mathbb{F})\times\mathrm{SL}(n,\mathbb{F})$ on $M(n,\mathbb{F})^{\oplus m}$ sending $(B_1,\ldots,B_m)$ to $(AB_1C^t,\ldots,AB_mC^t)$. This induces an action on the ring of polynomial functions on $M(n,\mathbb{F})^{\oplus m}$. Let $R(n,m)$ be the ring of those polynomials invariant under this action. $R(n,m)$ is called *the ring of matrix semi-invariants* (for m matrices of size $n\times n$) [24, 12]. The common zeros of all polynomials in $R(n,m)$, denoted as $N(R(n,m))$, is called the *nullcone* of this invariant ring in the invariant theory literature. The first link is the following result from invariant theory, proved using the celebrated Hilbert-Mumford criterion.

▶ **Theorem 2** ([4, 1]). *$(B_1,\ldots,B_m)$ is in $N(R(n,m))$ if and only if* $\mathrm{span}(B_1,\ldots,B_m)$ *has a shrunk subspace.*

Theorem 2 shows that matrix spaces with shrunk subspaces are characterized by those polynomials in $R(n,m)$. It is then desirable to know what polynomials in $R(n,m)$ look like. This task is usually resolved in the so-called first fundamental theorem for $R(n,m)$. In fact, it is this theorem that leads to the use of the blow-up operation.

▶ **Theorem 3** ([14, 31, 15, 1]). *Every homogeneous polynomial in $R(n,m)$ is of degree dn for some $d\in\mathbb{N}$, and is a linear combination of polynomials of the form* $\det(A_1\otimes X_1+\cdots+A_m\otimes X_m)$ *where X_i's are $n\times n$ variable matrices, and A_i's are $d\times d$ matrices over $\mathbb{F}$.*

Therefore, if $\mathrm{span}(B_1,\ldots,B_m)$ does not possess a shrunk subspace, then $(B_1,\ldots,B_m)$ is not in the null cone of $R(n,m)$ (Theorem 2). This implies that there exists some $(A_1,\ldots,A_m)\in M(d,\mathbb{F})^{\oplus m}$ such that $\det(A_1\otimes B_1+\cdots+A_m\otimes B_m)\neq 0$ (Theorem 3), which just says that $\mathcal{B}^{[d]}$ contains a non-singular matrix. This almost suggests that the blow-up operation would resolve the difficulty of certifying matrix spaces without shrunk subspaces, except that it is not immediately clear how large d needs to be for $\mathcal{B}^{[d]}$ to contain a non-singular matrix. To see that d is finite is classical: by Hilbert's basis theorem, $N(R(n,m))$ can be defined by finitely many polynomials, so d is also finite.

But if our hope is that the blow-up operation would *efficiently* certify matrix spaces without shrunk subspaces, just knowing d to be finite is not very useful – in fact, for this purpose we would require d to be upper bounded by a polynomial in n. Let $\sigma(R(n,m))$ be the smallest d such that $N(R(n,m))$ is defined by polynomials in $R(n,m)$ of degree no more than d. (This definition is valid for any invariant ring S, and to get an explicit upper bound on $\sigma(S)$ for invariant rings satisfying certain general conditions is an important research topic in invariant theory [28, 11].) Digging into the literature, from Derksen's work [11] it follows that $\sigma(R(n,m))\leq 1/4\cdot n^2\cdot 4^{n^2}$ for algebraically closed fields of characteristic 0.

Since 2015 there has been considerable progress towards a better upper bound on $\sigma(R(n,m))$. In [24] we show that $\sigma(R(n,m))\leq n!$ for all large enough fields. The key is the following so-called regularity lemma.

▶ **Lemma 4** (Regularity lemma for blow-ups, [24, Lemma 5.6]). *For $\mathcal{B} \leq M(n, \mathbb{F})$, assume that $|\mathbb{F}| > (nd)^{\Omega(1)}$. Then $\mathrm{crk}(\mathcal{B}^{[d]})$ is divisible by d.*

After [24] appeared, Derksen and Makam [12] discovered a concavity property of blow-ups, and showed that combining the regularity lemma with this property gives the following surprising result.

▶ **Theorem 5** ([12]). *Suppose $|\mathbb{F}| > n^{\Omega(1)}$. Then $\sigma(R(n,m)) \leq n^2 - n$.*

Theorem 5 immediately suggests an efficient approach to certify matrix spaces without shrunk subspaces: if $\mathcal{B}$ has no shrunk subspace, to see it we only need to exhibit a non-singular matrix in $\mathcal{B}^{[d]}$ for some $d \leq n-1$. A slightly more careful analysis suggests that if $\mathrm{ncrk}(\mathcal{B}) \geq r$, then to certify this we can exhibit a matrix in $\mathcal{B}^{[d]}$ of rank dr for some $d \leq r-1$.

By [11], this implies that $R(n,m)$ can be generated as a ring by those polynomials of degree $\leq O(n^8)$. Further consequences include improved degree bounds for generating semi-invariants of quivers (see [12]), since semi-invariants of quivers can be described as determinants of block matrices in general [15].

Soon after Derksen and Makam[12] announced their result, we discovered another argument based on the regularity lemma which gives $\sigma(R(n,m)) \leq n^2 + n$. While slightly worse in parameters, our argument is simpler than the one using the concavity property discovered by Derksen and Makam. We shall present this in Section 2.

1.4 Deterministic efficient algorithms for the non-commutative rank problem

As mentioned in Section 1.2, before 2015 it was only known that the non-commutative rank problem is in PSPACE. We gave a $\mathrm{poly}((n+1)!)$-time algorithm for this problem in 2015 [24]. Our algorithm can be viewed as a linear algebraic analogue of the augmenting path algorithm for the bipartite maximum matching problem, and relies heavily on a constructive proof of the regularity lemma.

Later that year Garg et al [20] showed that a careful analysis of an algorithm of Gurvits from 2003, [21], puts the non-commutative rank problem in P over $\mathbb{Q}$.

▶ **Theorem 6** ([20]). *Over $\mathbb{Q}$, the non-commutative rank problem is in* P.

Interestingly, Gurvits' algorithm is a quantum generalization of an algorithm for the bipartite maximum matching problem proposed by Linial et al,[26]: given the bipartite adjacency matrix of a bipartite graph, perform the row averaging and the column averaging operations alternatively. In [26] the authors proved that after a polynomially number of rounds, the resulting matrix is close to the identity matrix if and only if the bipartite graph has a perfect matching. Gurvits' algorithm is a beautiful and far-reaching generalization of the above idea to the setting of matrix spaces. It was originally used to tackle the commutative rank problem when the matrix space satisfies what Gurvits called the *Edmonds-Rado property*, namely those matrix spaces that are either non-singular, or have a shrunk subspace. The main innovation of Garg et al. in [20] is to realize that, by combining the exponential bounds on $\sigma(R(n,m))$ (either from [11] or [24]) with the capacity of operators introduced by Gurvits [21], Gurvits' algorithm actually solves the non-commutative rank problem over $\mathbb{Q}$. However, their algorithm fails to output a c-shrunk subspace and a matrix of rank $(n-c)d$ in $\mathcal{B}^{[d]}$ which together certify that the non-commutative rank is $n-c$.

We observed that Derksen and Makam's [12] concavity property of blow-ups can be constructivized, and this boosted our previous $\mathrm{poly}((n+1)!)$-time algorithm to polynomial

time. As mentioned later we discovered another, simpler, constructive argument, which also achieves the same bound. This leads to our main result.

▶ **Theorem 7** (Main theorem). *Let $\mathcal{B} \leq M(n, \mathbb{F})$ be a matrix space given by a linear basis, and suppose $|\mathbb{F}| = n^{\Omega(1)}$.*

Suppose that $\mathcal{B}$ has (a priori unknown) non-commutative rank r. Then there is a deterministic algorithm using $n^{O(1)}$ arithmetic operations over $\mathbb{F}$ that constructs a matrix of rank rd in a blow-up $\mathcal{B}^{[d]}$ for some $d \leq r + 1$ as well as an $(n - r)$-shrunk subspace of $\mathbb{F}^n$ for $\mathcal{B}$. When $\mathbb{F} = \mathbb{Q}$, the final data as well as all the intermediate data have size polynomial in the size of the input data and hence the algorithm runs in polynomial time.

Compared with the algorithm in [20], our algorithm has the advantages of working with arbitrary large enough fields, and outputting a shrunk subspace and a matrix in a blow-up space certifying that the non-commutative rank is r. Note that the second feature is new even over $\mathbb{Q}$. In section 1.5 we show that the small finite fields case can also be handled.

▶ Remark.

(a) If the constructivized version of Derksen and Makam [12] is used (see Appendix A), then d in the above theorem can be improved to $d \leq r - 1$ instead of $d \leq r + 1$.

(b) Polynomial running time of the algorithm can also be proved for a wide range "concrete" base fields $\mathbb{F}$. These include sufficiently large finite fields, and also number fields and transcendental extensions of constant degree over finite fields and over number fields.

(c) In particular, the non-commutative rank can be computed in deterministic polynomial time in positive characteristic as well, assuming that the ground field is sufficiently large.

Our result also settles a question of Gurvits [21], asking if it is possible to decide efficiently, over fields of positive characteristics, whether or not there exists a non-singular matrix in a matrix space having the Edmonds-Rado property. Since the algorithm in Theorem 7 efficiently tells whether the given matrix space has a shrunk subspace (e.g. the non-commutative rank is not full), it settles Gurvits' question, when the field size is as stated in the hypothesis.

1.5 Over small finite fields

From the above, we have seen a polynomial upper bound on $\sigma(R(n, m))$, and settled the non-commutative rank problem as well as SDIT for the Edmonds-Rado class, provided that the underlying field is large enough. However we can say more, even when the base field is a "too small" finite field.

▶ **Corollary 8.** *Let $\mathbb{F}$ be a finite field of size $s < n^{O(1)}$.*

1. *Let $R(n, m)$ be the ring of matrix semi-invariants over $\mathbb{F}$. Then $\sigma(R(n, m)) \leq O((n^2 - n) \log_s n)$.*
2. *Let $\mathcal{B} \leq M(n, \mathbb{F})$ be a matrix space given by a linear basis with a priori unknown non-commutative rank r. There is a deterministic polynomial-time algorithm that constructs a matrix of rank rd in a blow-up $\mathcal{B}^{[d]}$ for some $d \leq O(r \log_s n)$, as well as an $(n - r)$-shrunk subspace of $\mathbb{F}^n$ for $\mathcal{B}$.*
3. *Let $\mathcal{B} \leq M(n, \mathbb{F})$ be a matrix space given by a linear basis satisfying the Edmonds-Rado property. Then there exists a deterministic polynomial-time algorithm that decide whether $\mathcal{B}$ has a non-singular matrix, or a shrunk-subspace.*

1.6 A technical improvement of the regularity lemma

As mentioned in Section 1.3, the regularity lemma from [24] (Lemma 4) is the key to the polynomial bound on $\sigma(R(n, m))$. Furthermore, a constructive version of it from [24] is crucial

for our algorithm for the non-commutative rank problem. Specifically, we use the regularity lemma in the algorithm in the following situation: given $A \in \mathcal{B}^{[d]}$ of rank $(r-1)d+k$ where $1 < k < d$, we wish to construct $A' \in \mathcal{B}^{[d]}$ of rank $\geq rd$ efficiently. In [24, Lemma 5.7], this was achieved under the condition that, if $\mathrm{char}(\mathbb{F}) = p > 0$, then $p \nmid d$. In this paper, we remove this coprime condition. Roughly speaking, the proof of the regularity lemma in [24] made crucial use of central division algebras, and a classical construction of such algebras is based on cyclic field extensions. Therefore an efficient construction of cyclic field extensions is the basis of the constructive regularity lemma. In [24] an efficient construction of cyclic field extensions is presented assuming that the extension degree and the field characteristic are coprime. We remove this coprime condition to get a complete constructive proof of the regularity lemma. The technical details of this construction are given in Appendix B.

We note that recently Derksen and Makam obtained another proof of the regularity lemma in [13]. However they remark [13] that their proof is "less constructive".

Organization of the article

In Section 2 we present our proof of the n^2+n upper bound on $\sigma(R(n,m))$. In Section 3 we outline the algorithm in [24] and explain how the idea in Section 2 can be used to bring down its time complexity from exponential to polynomial, both intuitively (Section 3.1) and rigorously (Section 3.2). Section 4 contains the proof of Corollary 8. A proof of the full constructive regularity lemma is given in section 5. In Appendix A we show how the concavity property of Derksen and Makam can be constructivized. In Appendix B, we give an efficient construction of cyclic field extensions.

2 Another proof of a polynomial upper bound on defining the nullcone of matrix semi-invariants

In this short section we show how a simple argument based on the regularity lemma (Lemma 4) proves that $\sigma(R(n,m)) \leq n^2+n$. Let us remark again that Derksen and Makam [12] were the first to prove that $\sigma(R(n,m)) \leq n^2-n$. They too relied crucially on the regularity lemma from [24]. Here we present a simpler proof, albeit with a slightly worse parameter.

▶ **Theorem 9.** *Suppose* $|\mathbb{F}| > n^{\Omega(1)}$. *Then* $\sigma(R(n,m)) \leq n^2+n$.

Proof. Suppose $\mathcal{B} \leq M(n,\mathbb{F})$ is of dimension m. To prove the statement, we need to show that for any such $\mathcal{B}$, $\mathcal{B}^{[d]}$ contains a non-singular matrix for some $d \leq n+1$. As explained in Section 2, by Theorem 2 and 3, as well as Hilbert's basis theorem, $\mathcal{B}^{[d]}$ contains a non-singular matrix A for some $d \in \mathbb{N}$. If $d > n+1$, then note that $A \in \mathcal{B}^{[d]} \leq M(d,\mathbb{F}) \otimes M(n,\mathbb{F})$ is a $d \times d$ block matrix where each block is of size $n \times n$. Let A' be the right lower $(d-1) \times (d-1)$ block matrix of A; note that $A' \in \mathcal{B}^{[d-1]}$. Since A' is obtained from A by removing n rows and n columns, we have $\mathrm{rk}(A') \geq dn-2n = dn-n+1-(n+1) > dn-n+1-d = (d-1)(n-1)$, which gives $\mathrm{crk}(\mathcal{B}^{[d-1]}) > (d-1)(n-1)$. By Lemma 4, $\mathrm{crk}(\mathcal{B}^{[d-1]})$ has to be divisible by $(d-1)$, so $\mathrm{crk}(\mathcal{B}^{[d-1]}) = (d-1)n$ is of full rank. Continuing this way we can reduce d to be no more than $n+1$. ◀

3 Proof of the main theorem

As mentioned in Section 1.4, the algorithm for Theorem 7 is obtained by combining the algorithm in [24] with the idea in the proof of Theorem 9. For the readers convenience we give an intuitive explanation of the algorithm from [24] in Section 3.1; the reader is referred

to [24] for a rigorous treatment. We also explain why it runs in exponential time and how the idea in the proof of Theorem 9 can reduce its complexity to polynomial. Section 3.2 gives a rigorous treatment of the algorithm that proves Theorem 9.

3.1 Outline of the algorithm in [24]

The algorithm in [24] can be viewed as an analogue of the augmenting path algorithm for the bipartite matching problem. However, due to the failure of the analogue of Hall's marriage theorem in the matrix space setting, there are a couple of new and sophisticated components.

Given a matching T for the input bipartite graph $G = (L \cup R, E)$, the algorithm tries to find an augmenting path for T. If an augmenting path is found, T is replaced by a larger matching T'. If no augmenting paths can be found, the algorithm can output a shrunk subset as the certificate of the maximality of T.

We hope to implement the above idea for the non-commutative rank problem. Given a matrix $A \in \mathcal{B} = \mathrm{span}(B_1, \ldots, B_m) \leq M(n, \mathbb{F})$, we would like to either find an "augmenting path" for it and increase its rank, or output a c-shrunk subspace where $c = \mathrm{cork}(A)$.

A linear algebraic analogue of augmenting paths has been developed in [23]. Given a subspace $U \leq \mathbb{F}^n$, let $A^{-1}(U)$ be the preimage of U under A, namely the subspace $\{v \in \mathbb{F}^n : A(v) \in U\}$. We also define $\mathcal{B}(U) := \mathrm{span}(\cup_{i\in[m]} B_i(U))$. Given $A \in \mathcal{B} \leq M(n, \mathbb{F})$, we apply $\mathcal{B}$ and A^{-1} iteratively to $V_0 = \ker(A)$, to get $W_1 = \mathcal{B}(V_0)$, $V_1 = A^{-1}(W_1)$, $W_2 = \mathcal{B}(V_1)$, $\ldots$, $V_i = A^{-1}(W_i)$, $W_{i+1} = \mathcal{B}(V_i)$, $\ldots$. It can be shown that for some $\ell \in [n]$, $W_1 < W_2 < \cdots < W_\ell = W_{\ell+1} = \ldots$, and this is called *the second Wong sequence* of $(A, \mathcal{B})$.[5][6] W_ℓ is called the *limit subspace* of this sequence.

In [23], it was proved that $W_\ell \leq \mathrm{im}(A)$ if and only if $\mathcal{B}$ has a $\mathrm{cork}(A)$-shrunk subspace[7]. Therefore when $W_\ell \leq \mathrm{im}(A)$, we can conclude that the non-commutative rank is $\mathrm{rk}(A)$. On the other hand, when $W_\ell \not\leq \mathrm{im}(A)$, following the bipartite maximum matching algorithm it seems natural to try to obtain $A' \in \mathcal{B}$ with $\mathrm{rk}(A') > \mathrm{rk}(A)$. However this is not always possible, as it can be the case that $\mathrm{rk}(A) = \mathrm{crk}(\mathcal{B})$ and $\mathrm{crk}(\mathcal{B}) < \mathrm{ncrk}(\mathcal{B})$. Thanks to Theorem 9, the key to resolve this difficulty is to find $A' \in \mathcal{B}^{[d]}$ of rank $\geq (r+1)d$ for some not too large d. (So that the scaled-down rank $\mathrm{rk}(A')/d$ is larger than r.) This is accomplished in two steps.

The first step is to obtain a matrix $\widehat{A} \in \mathcal{B}^{[d]}$ of rank $\geq rd + 1$ where $d = r + 1$. To see how this step works, notice first that by multiplying A and $\mathcal{B}$ with an appropriate matrix, one can arrange A to be idempotent. In that case, as long as $W_1, \ldots, W_{j-1}$ remain inside $\mathrm{im}(A)$, we have $W_j = \mathcal{B}^j \ker(A)$. Let k be the smallest index j with $W_j \not\leq \mathrm{im}(A)$. Obviously $k \leq r + 1$. Then there exist indices $1 \leq i_1, \ldots, i_k \leq m$ such that $B_1 \cdots B_k \ker(A) \not\leq \mathrm{im}(A)$. It would be nice if one could find a *single* matrix $B \in \mathcal{B}$ such that $B^k \ker(A) \not\leq \mathrm{im}(A)$: if this indeed happens, then for some λ and μ from a subset of the base field of size at least $r + 1$ one would have $\mathrm{rk}(\mu A + \lambda B) > \mathrm{rk}(A)$. This is because of the result from [3], where it is proved that for two-dimensional matrix spaces the commutative and the non-commutative ranks coincide.

---

[5] The first Wong sequence is the dual of the second one. This naming convention is due to Wong [34] who defined the two sequences for the special case when $\mathcal{B}$ is of dimension 1. See [23] for more details.

[6] Over $\mathbb{Q}$ the straightforward implementation of the second Wong sequence may lead to a bit size explosion. To avoid that some tricks are needed; see [23].

[7] At the time of writing the first version of [23], the authors were unaware of [19] where actually this statement has already appeared. The real value added in [23] was that, in certain special cases, when $W_\ell \not\leq \mathrm{im}(A)$ the second Wong sequence could be used to find an "augmenting" matrix B from $\mathcal{B}$ such that $\mathrm{rk}(\mu A + \lambda B) > \mathrm{rk}(A)$ for some scalars λ and μ.

The main ingredient of the algorithm in [23] was a method to find such a $B \in \mathcal{B}$ in certain special cases. The idea in [24] is that if we relax our requirement and instead work in $\mathcal{B}^{[d]}$, then this can be achieved in a simple way, for every matrix space $\mathcal{B}$. Observe that $A \otimes I_d$ is a matrix from $\mathcal{B}^{[d]}$ of rank rd. Let $E_{i,j}$ be the elementary matrix with the (i,j)th entry being 1 and others 0. Put $\widehat{B} = B_1 \otimes E_{1,2} + B_2 \otimes E_{2,3} + \ldots B_{k-1} \otimes E_{k-1,k} + B_k \otimes E_{k,1} \in \mathcal{B}^{[d]}$. Then $\widehat{B}^k = (B_1 \cdots B_k) \otimes E_{1,1} + (B_2 \cdots B_k B_1) \otimes E_{2,2} + \ldots + (B_k B_1 \cdots B_{k-1}) \otimes E_{k,k}$, whence $\widehat{B}^k \ker(A \otimes I_d) \not\leq \mathrm{im}(A \otimes I_d)$. Therefore $\widehat{A} = \mu(A \otimes I_d) + \lambda \widehat{B}$ will have rank larger than rd for some λ and μ from a subset of the base field of size $rd + 1$.

For the second step, starting with $\widehat{A}$, we wish to get the desired $A' \in \mathcal{B}^{[d]}$ of rank $\geq (r+1)d$. This is accomplished by the constructive regularity lemma ([24, Lemma 5.7]; see Lemma 11 in Section 3.2).

This $A' \in \mathcal{B}^{[d]}$ of rank $\geq (r+1)d$ where $d = r + 1$ certifies that $\mathrm{ncrk}(\mathcal{B}) \geq r + 1$. (If $\mathrm{ncrk}(\mathcal{B}) \leq r$ then $\mathrm{crk}(\mathcal{B}^{[d]}) \leq rd$ for any d by Proposition 1.) So after these two steps we obtain A' of rank $r'd$ where $r' > r$.

In the next phase, we need to use A' and $\mathcal{B}^{[d]}$ to restart the above procedure, hoping either to find a $\mathrm{cork}(A')$-shrunk subspace, or to obtain some A'' in $\mathcal{B}^{[dd']}$ of rank $r''dd'$ where $r'' > r'$. We then apply the second Wong sequence to work with the blow-up space $\mathcal{B}^{[d]}$ and A'.[8] If $\mathrm{cork}(A')$-shrunk subspace U' is found for $\mathcal{B}^{[d]}$, then this naturally induces a $\mathrm{cork}(A')/d$-shrunk subspace U for $\mathcal{B}$ [24, Proposition 5.2]. In this case we conclude that the non-commutative rank is r', and A' and U together serve as witnesses for this fact. If the limit subspace goes out of $\mathrm{im}(A')$ we need to go to an even larger blow-up space $(\mathcal{B}^{[d]})^{[d']} \cong \mathcal{B}^{[dd']}$ where $d' = r' + 1$ to find a matrix $A'' \in \mathcal{B}^{[dd']}$ of rank $r''dd'$ for some $r'' > r'$.

So the right approach to carrying out the augmenting path idea in this setting is to play with shrunk subspaces on one hand, and matrices in the blow-up spaces on the other.

The alert reader may now notice that the above strategy leads to an exponential-time algorithm. Recall that we start with $A \in \mathcal{B}$ of rank r. If $\mathrm{ncrk}(\mathcal{B}) = n$, then we may end up finding $A^* \in \mathcal{B}^{[d^*]}$ of rank nd^* where d^* can be as large as $n!/r!$. This is because, increasing the scaled-down rank from r' to $r' + 1$ would lead to a multiplicative factor of $r' + 1$ in the size of the blow-up space. This is why the algorithm in [24] runs in time $\mathrm{poly}(n!)$.

However, the idea in the proof of Theorem 9 readily implies that, once we find A' of rank $r'd$ in $\mathcal{B}^{[d]}$, we can efficiently reduce d to be no more than $r' + 1$. This means that we can always ensure that the blow-up factor is small, which is the key to reducing the complexity from exponential time to polynomial time. We make the above idea rigorous.

3.2 The algorithm for the main theorem

In this subsection we prove Theorem 7. Here it is easier to work with $\mathcal{B}^{\{d\}} := \mathcal{B} \otimes M(d, \mathbb{F})$ instead of $\mathcal{B}^{[d]} = M(d, \mathbb{F}) \otimes \mathcal{B}$. This does not change anything, as $\mathcal{B}^{[d]}$ is isomorphic to $\mathcal{B}^{\{d\}}$. The point is that we now think of matrices in the blow-up space as $n \times n$ block matrices with blocks of size $d \times d$. We first recall some notions from [24].

Finding an sd-shrunk subspace for the $\mathcal{B}^{\{d\}}$ is equivalent to finding an s-shrunk subspace for $\mathcal{B}$ because of the following simple observations ([24, Proposition 5.2]). Firstly, for every s-shrunk subspace U of $\mathbb{F}^n$ the subspace $U \otimes \mathbb{F}^d$ for $\mathcal{B}$ is an sd-shrunk subspace for $\mathcal{B}^{\{d\}}$. Conversely, a s'-shrunk subspace for $\mathcal{B}^{\{d\}}$ can be embedded into a subspace of the form $U \otimes \mathbb{F}^d$ where U is an s-shrunk subspace for $\mathcal{B}$ with $sd \geq s'$.

[8] When the second Wong sequence is applied to such blow-up spaces then it has some nice properties; cf. the proof for Theorem 5.10 in [24].

We shall also need the following useful lemma.

▶ **Lemma 10** (Data reduction, [10] and [24, Lemma 5.3]). *Let $\mathcal{B} \leq M(k \times \ell, \mathbb{F})$ be given by a basis $B_1, \ldots, B_m$, and let $\mathbb{K}$ be an extension field of $\mathbb{F}$. Let S be a subset of $\mathbb{F}$ of size at least $r+1$. Suppose that we are given a matrix $A' = \sum_i a'_i B_i \in \mathcal{B} \otimes_{\mathbb{F}} \mathbb{K}$ of rank at least r. Then we can find $A = \sum_i a_i B_i$ of rank also at least r with $a_i \in S$. The algorithm uses* $\mathrm{poly}(k, \ell, r)$ *rank computations for matrices of the form $\sum \sigma_i A_i$ where $a''_i \in \{a'_1, \ldots, a'_m\} \cup S$.*

We now present the formal statement of the constructive regularity lemma in its full generality, with an addition of a technical notion that will be useful for the proof of Theorem 7. Let $n \in \mathbb{N}$, and let $\mathbf{i} = (i_1, \ldots, i_r)$, $\mathbf{j} = (j_1, \ldots, j_r)$ be two sequences of integers, where $1 \leq i_1 < \cdots < i_r \leq n$ and $1 \leq j_1 < \cdots < j_r \leq n$. For a matrix $A \in M(n, \mathbb{F}) \otimes M(d, \mathbb{F})$, the $r \times r$ *window* indexed by $\mathbf{i}, \mathbf{j}$ is the sub-matrix of A consisting of the blocks indexed by (i_k, j_ℓ), $k, \ell \in [r]$.

▶ **Lemma 11** (The complete constructive regularity lemma). *For $\mathcal{B} \leq M(n, \mathbb{F})$ and $\mathcal{A} = \mathcal{B}^{\{d\}}$, assume that $|\mathbb{F}| = (rd)^{\Omega(1)}$. Given a matrix $A \in \mathcal{A}$ with $\mathrm{rk} A > (r-1)d$, there exists a deterministic algorithm that returns $\widetilde{A} \in \mathcal{A}$ and an $r \times r$ window W in $\widetilde{A}$ such that W is non-singular (of rank rd). This algorithm uses* $\mathrm{poly}(nd)$ *arithmetic operations and, over $\mathbb{Q}$, the algorithm runs in polynomial time (in particular, all intermediate numbers have bit lengths polynomial in the input size).*

The cases (a) $\mathrm{char}(\mathbb{F}) = 0$, (b) $\mathrm{char}(\mathbb{F})$ and d are coprime, and $|\mathbb{F}| = (rd)^{\Omega(1)}$ were settled in [24, Lemma 5.7]. We will remove this coprime condition in .

The main technical ingredient of our algorithm will be the following result,[24, Theorem 5.10] based on Lemma 11, and the second Wong sequences introduced in [19] and [23]. It states that either a shrunk subspace witnessing that the (scaled-down) rank of a matrix in a blow-up reaches the non-commutative rank or a matrix in a larger blow-up having larger scaled-down rank can be efficiently constructed.

▶ **Theorem 12** ([24, Theorem 5.10]). *Let $\mathcal{B} \leq M(n, \mathbb{F})$ and let $\mathcal{A} = \mathcal{B}^{\{d\}}$. Assume that we are given a matrix $A \in \mathcal{A}$ with $\mathrm{rk}(A) = rd$, and $|\mathbb{F}|$ is $(ndd')^{\Omega(1)}$, where $d' > r$ is any positive integer.*

There exists a deterministic algorithm that returns either an $(n-r)d$-shrunk subspace for $\mathcal{A}$ (equivalently, an $(n-r)$-shrunk subspace for $\mathcal{B}$), or a matrix $B \in \mathcal{A} \otimes M(d', \mathbb{F})$ of rank at least $(r+1)dd'$. Furthermore, in the latter case an $(r+1) \times (r+1)$ window is also found such that the corresponding $(r+1)dd' \times (r+1)dd'$ sub-matrix of B has full rank. This algorithm uses $\mathrm{poly}(ndd')$ *arithmetic operations and, over $\mathbb{Q}$, all intermediate numbers have bit lengths polynomial in the input size.*

The sentence on the $(r+1) \times (r+1)$ window was not explicitly stated in [24]. However, the algorithm in its proof contains, as a last step, a call to the method behind Lemma 11. Also, the theorem was stated only under the assumption that d was not divisible by $\mathrm{char}(\mathbb{F})$ because of this last call. As the algorithm up to this step constructs a matrix of rank greater than rdd', the complete constructive regularity lemma, as stated in Lemma 11, makes it possible to dispense with that assumption.

To complete the proof Theorem 7, the regularity lemma needs to be accompanied with a reduction procedure that keeps the blow-up parameter small. In this section we use our method, which follows immediately from the proof of Theorem 9. The method based on the Derksen-Makam idea is presented in Appendix A.

▶ **Lemma 13.** *Let $\mathcal{B} \leq M(n,\mathbb{F})$, and $d > n+1$. Assume we are given a matrix $A \in \mathcal{B}^{\{d\}}$ of rank dn. Then there exists a deterministic polynomial-time procedure that constructs $A' \in \mathcal{B}^{\{d-1\}}$ of rank $(d-1)n$.*

Proof. Let A'' be an appropriate $(d-1)n \times (d-1)n$ submatrix of A corresponding to a matrix in $\mathcal{B}^{\{d-1\}}$. We claim A'' is of rank $> (d-1)(n-1)$. Suppose not, as A is obtained from A'' from adding n rows and then n columns, and $d > n+1$, we have $\mathrm{rk}(A) \leq \mathrm{rk}(A'') + 2n \leq dn - d - n + 1 + 2n < dn$, a contradiction. Now that $\mathrm{rk}(A'') > (d-1)(n-1)$, using Lemma 11, we obtain $A' \leq \mathcal{B}^{\{d-1\}}$ of rank $(d-1)n$. ◀

Proof of Theorem 7. Let $B_1, \ldots, B_m$ be the input basis for $\mathcal{B}$. The algorithm is an iteration based on Theorem 12. In each round we start with a matrix $A = \sum_i B_i \otimes T_i \in \mathcal{B}^{\{d\}}$ of rank rd for some integer $d \leq r+1$.

In the first round, $d = 1$ and A can be taken as any matrix in $\mathcal{B}$. The procedure behind Theorem 12 either returns an $(n-r)$-shrunk subspace (in which case we are done), or a new matrix (denoted also by A) in a blow-up $\mathcal{B}^{\{d'\}}$ of rank $\geq (r+1)d'$ for some $d' \leq (r+1)^2$, together with a square window of size $r+1$ so that the corresponding sub-matrix of A is of rank $(r+1)d'$.

If $d' > r+2$ then we apply Lemma 13 as follows. The n in the statement of Lemma 13 will be $r+1$, and we use it repeatedly to get a matrix in the $(r+2)$-blow-up, the main content of which consists of $(r+2) \times (r+2)$ matrices $T'_1, \ldots, T'_m$ such that the corresponding $(r+1)(r+2) \times (r+1)(r+2)$ sub-matrix of $A' = \sum_i B_i \otimes T'_i$ has full rank. Then we replace A with A' and apply the size reduction procedure in Lemma 10 to ensure that the entries of T_i are from the prescribed subset of $\mathbb{F}$, and continue the iteration with this new matrix A. ◀

4 Proof of Corollary 8: the case of small finite fields

We only need to prove Corollary 8 (2), from which (1) and (3) are immediate.

Given a matrix space $\mathcal{B} \leq M(n,\mathbb{F})$ and a field extension $\mathbb{K}/\mathbb{F}$, $\mathcal{B}$ can be viewed naturally as a matrix space in $M(n,\mathbb{K})$. For convenience we use $\mathrm{ncrk}_{\mathbb{F}}(\mathcal{B})$ to signal that we consider the non-commutative rank of $\mathcal{B}$ over $\mathbb{F}$. Observe first that the non-commutative rank does not change under field extensions. This is classical, and can be seen from the perspective of the second Wong sequences (see e.g. [23, Section 2]). Note that unlike the non-commutative rank the commutative rank may get larger if we go to an extension field of a too-small field.

▶ **Lemma 14.** *Given $\mathcal{B} \leq M(n,\mathbb{F})$ and a field extension $\mathbb{K}/\mathbb{F}$, we have $\mathrm{ncrk}_{\mathbb{F}}(\mathcal{B}) = \mathrm{ncrk}_{\mathbb{K}}(\mathcal{B})$.*

Suppose $\mathcal{B} \leq M(n,\mathbb{F})$ is given by a linear basis $\{B_1, \ldots, B_m\}$. Let $\mathbb{K}/\mathbb{F}$ be a field extension of degree g so that $|\mathbb{K}| = n^{\Omega(1)}$ satisfies the field size condition of Theorem 7. Note that $g \leq O(\log_{|\mathbb{F}|} n)$. Viewing $\mathcal{B}$ as a matrix space over $\mathbb{K}$, we apply Theorem 7 to compute $\mathrm{ncrk}_{\mathbb{K}}(\mathcal{B})$, which is equal to $r = \mathrm{ncrk}_{\mathbb{F}}(\mathcal{B})$ by Lemma 14. We also obtain the following: (1) $A_1, \ldots, A_m \in M(d,\mathbb{K})$ such that $A = \sum_{i\in[m]} A_i \otimes B_i$ is of rank rd, and (2) $U \leq \mathbb{K}^n$ such that U is a shrunk subspace of $\mathcal{B}$ a matrix space in $M(n,\mathbb{K})$. We fix an embedding ϕ of $\mathbb{K}$ into $M(g,\mathbb{F})$ using the regular representation. For $i \in [m]$, construct $\widetilde{A_i} \in M(gd,\mathbb{F})$ by replacing each entry α of A_i with $\phi(\alpha)$, and form $\widetilde{A} = \sum_{i\in[m]} \widetilde{A_i} \otimes B_i$. Note that $\widetilde{A}$ is in $M(gd,\mathbb{F}) \otimes \mathcal{B}$, and it can be seen easily that $\mathrm{rk}(\widetilde{A}) = g \cdot \mathrm{rk}(A)$. Since $\mathrm{rk}(\widetilde{A})/gd = r = \mathrm{ncrk}_{\mathbb{F}}(\mathcal{B})$, we have $\mathrm{crk}_{\mathbb{F}}(M(gd,\mathbb{F}) \otimes \mathcal{B}) = \mathrm{ncrk}_{\mathbb{F}}(M(gd,\mathbb{F}) \otimes \mathcal{B})$. This implies that we can apply the second Wong sequence to $(\widetilde{A}, M(gd,\mathbb{F}) \otimes \mathcal{B})$ to obtain an $(n-r)gd$-shrunk subspace of $M(gd,\mathbb{F}) \otimes \mathcal{B}$ which then induces an $(n-r)$-shrunk subspace of $\mathcal{B}$.

5 The full constructive regularity lemma

In this section we prove Lemma 11. The proof makes use of the following two results from [24], which we reproduce here as Proposition 15 and Lemma 16. Roughly speaking, Proposition 15 states that from cyclic field extensions one can construct central division algebras. Lemma 16 is the conditional constructive regularity lemma, where the condition required there is an efficient construction of central division algebras.

▶ **Proposition 15** ([24, Proposition 4.4]). *Let $\mathbb{L}$ be a cyclic extension of degree d of a field $\mathbb{K}$, and suppose that $\mathbb{L}$ is given by structure constants w.r.t. a $\mathbb{K}$-basis $A_1, \ldots, A_d$. Similarly, a generator σ for the Galois group is assumed to be given by its matrix in terms of the same basis. Let Y be a formal variable. Then one can construct a $\mathbb{K}(Y)$-basis σ of $M(d, \mathbb{K}(Y))$ such that the $\mathbb{K}(Y^d)$-linear span of σ is a central division algebra over $\mathbb{K}(Y^d)$ of index d, using* $\mathrm{poly}(d)$ *arithmetic operations in $\mathbb{K}$. Furthermore for $\mathbb{K} = \mathbb{Q}[\sqrt[d]{1}]$, the bit complexity of the algorithm, as well as the size of the output, are also* $\mathrm{poly}(d)$.

▶ **Lemma 16** (Conditional regularity [24, Lemma 5.4]). *Assume that we are given a matrix $A \in \mathcal{B}^{\{d\}} \leq M(dn, \mathbb{F})$ with $\mathrm{rk}(A) = (r-1)d + k$ for some $1 < k < d$. Let X and Y be formal variables and put $\mathbb{K} = \mathbb{F}'(X)$, where $\mathbb{F}'$ is a finite extension of $\mathbb{F}$ of degree at most d. Suppose further that $|\mathbb{F}| > (nd)^{O(1)}$ and that we are also given a $\mathbb{K}(Y)$-basis σ of $M(d, \mathbb{K}(Y))$ such that the $\mathbb{K}(Y^d)$-linear span of σ is a central division algebra D' over $\mathbb{K}(Y^d)$. Let δ be the maximum of the degrees of the polynomials appearing as numerators or denominators of the entries of the matrices in σ. Then, using $(nd + \delta)^{O(1)}$ arithmetic operations in $\mathbb{F}$, one can find a matrix $A'' \in \mathcal{B}^{\{d\}}$ with $\mathrm{rk}(A'') \geq rd$. Furthermore, over $\mathbb{Q}$ the bit complexity of the algorithm is polynomial in the size of the input data (that is, the total number of bits describing the entries of matrices and in the coefficients of polynomials).*

From the above two results, the missing piece is just an efficient construction of cyclic field extensions. In [24] such a construction based on Kummer extensions was presented assuming $\mathrm{char}(\mathbb{F})$ and d are coprime. The main issue with the case when d is divisible by $\mathrm{char}(\mathbb{F})$ is that the proof requires an efficient construction of an appropriate Artin-Schreier-Witt extension of $\mathbb{F}_p(x)$, not known to us when writing [24]. Now we have such a construction leading to the following lemma whose proof we give in appendix B.

▶ **Lemma 17.** *Let $\mathbb{F}'$ be a field. Let d be any non-negative integer. If $\mathrm{char}(\mathbb{F}') = 0$ then $d_1 = d$. If $\mathrm{char}(\mathbb{F}') = p > 0$ then let d_1 be the p-free part of d, that is, $d = d_1 p^s$, where $p \nmid d_1$ and $s \in \mathbb{N}$.*

Assume that $\mathbb{F}'$ contains a known d_1th root of unity ζ. Then a cyclic extension $\mathbb{L}$ degree d of $\mathbb{K} := \mathbb{F}'(X)$ can be computed using $\mathrm{poly}(d)$ *arithmetic operations. $\mathbb{L}$ will be given by structure constants with respect to a basis, and the matrix for a generator of the Galois group in terms of the same basis will also be given. All the output entries (the structure constants as well as the entries of the matrix representing the Galois group generator) will be polynomials of degree* $\mathrm{poly}(d)$ *in $\mathbb{F}'[X]$. Furthermore for $\mathbb{F}' = \mathbb{Q}[\sqrt[d_1]{1}]$, the bit complexity of the algorithm (as well as the size of the output) is* $\mathrm{poly}(d)$.

Proof of Lemma 11. The statement, except the window part, readily follows by plugging Lemma 17 and Proposition 15 to Lemma 16.

To see that such a window can be computed, we first observe that the lemma applies to d-blow-ups of rectangular matrices, by simple zero padding. Second, apply the lemma and find an $rd \times rd$ nonsingular sub-matrix of the given matrix A. If the column indices include some such that not all of its $d-1$ siblings are included, then (1) delete the corresponding

column from the original matrix space; (2) let A' be the matrix obtained by deleting the corresponding d columns from A. Then $\mathrm{rk}(A') > \mathrm{rk}(A) - (d-1)$. So we apply the regularity lemma in the rectangular space with A', to round up the rank to $\mathrm{rk}(A)$ again. Do the same for row indices. Iterate until we obtain a full window. ◀

Acknowledgements. We would like to thank the authors of [20] and of [12] for sharing their ideas with us and making it possible for us to read early versions of their manuscripts. Part of the work was done when Gábor and Youming were visiting the Centre for Quantum Technologies at the National University of Singapore.

References

1 B. Adsul, S. Nayak, and K. V. Subrahmanyam. A geometric approach to the Kronecker problem II: rectangular shapes, invariants of matrices and the Artin–Procesi theorem. preprint, 2007.

2 M. D. Atkinson. Primitive spaces of matrices of bounded rank. II. *Journal of the Australian Mathematical Society (Series A)*, 34(03):306–315, 1983.

3 MD Atkinson and S Lloyd. Primitive spaces of matrices of bounded rank. *Journal of the Australian Mathematical Society (Series A)*, 30(04):473–482, 1981.

4 M. Bürgin and J. Draisma. The Hilbert null-cone on tuples of matrices and bilinear forms. *Mathematische Zeitschrift*, 254(4):785–809, 2006.

5 Marco Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Tighter connections between derandomization and circuit lower bounds. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, 2015, Princeton, NJ, USA*, pages 645–658, 2015. doi:10.4230/LIPIcs.APPROX-RANDOM.2015.645.

6 P. M. Cohn. The word problem for free fields. *J. Symbolic Logic*, 38(2):309–314, 06 1973. URL: http://projecteuclid.org/euclid.jsl/1183738636.

7 P. M. Cohn. The word problem for free fields: A correction and an addendum. *J. Symbolic Logic*, 40(1):69–74, 03 1975. URL: http://projecteuclid.org/euclid.jsl/1183739310.

8 P. M. Cohn. *Skew Fields: Theory of General Division Rings*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1995. URL: http://books.google.com.au/books?id=u-4ADgUgpSMC.

9 P. M. Cohn and C. Reutenauer. On the construction of the free field. *International Journal of Algebra and Computation*, 9(3-4):307–323, 1999.

10 Willem A. de Graaf, Gábor Ivanyos, and Lajos Rónyai. Computing Cartan subalgebras of Lie algebras. *Applicable Algebra in Engineering, Communication and Computing*, 7(5):339–349, 1996.

11 Harm Derksen. Polynomial bounds for rings of invariants. *Proceedings of the American Mathematical Society*, 129(4):955–964, 2001.

12 Harm Derksen and Visu Makam. Polynomial degree bounds for matrix semi-invariants. preprint ArXiv:1512.03393, 2015.

13 Harm Derksen and Visu Makam. On non-commutative rank and tensor rank. Preprint ArXiv:1606.06701, 2016.

14 Harm Derksen and Jerzy Weyman. Semi-invariants of quivers and saturation for littlewood-richardson coefficients. *Journal of the American Mathematical Society*, 13(3):467–479, 2000.

15 M. Domokos and A. N. Zubkov. Semi-invariants of quivers as determinants. *Transformation groups*, 6(1):9–24, 2001.

16 Jan Draisma. Small maximal spaces of non-invertible matrices. *Bulletin of the London Mathematical Society*, 38:764–776, 10 2006. doi:10.1112/S0024609306018741.

17 Jack Edmonds. Systems of distinct representatives and linear algebra. *J. Res. Nat. Bur. Standards Sect. B*, 71:241–245, 1967.

18 David Eisenbud and Joe Harris. Vector spaces of matrices of low rank. *Advances in Mathematics*, 70(2):135 – 155, 1988. doi:10.1016/0001-8708(88)90054-0.

19 M. Fortin and C. Reutenauer. Commutative/noncommutative rank of linear matrices and subspaces of matrices of low rank. *Séminaire Lotharingien de Combinatoire*, 52:B52f, 2004.

20 Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. A deterministic polynomial time algorithm for non-commutative rational identity testing. Preprint ArXiv:1511.03730. To appear in FOCS 2016, 2015.

21 Leonid Gurvits. Classical complexity and quantum entanglement. *J. Comput. Syst. Sci.*, 69(3):448–484, 2004.

22 Pavel Hrubeš and Avi Wigderson. Non-commutative arithmetic circuits with division. *Theory of Computing*, 11:357–393, 2015. doi:10.4086/toc.2015.v011a014.

23 Gábor Ivanyos, Marek Karpinski, Youming Qiao, and Miklos Santha. Generalized wong sequences and their applications to edmonds' problems. *J. Comput. Syst. Sci.*, 81(7):1373–1386, 2015. doi:10.1016/j.jcss.2015.04.006.

24 Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Non-commutative Edmonds' problem and matrix semi-invariants. Preprint arXiv:1508.00690, to appear in Computational Complexity, doi:10.1007/s00037-016-0143-x, 2015. doi:10.1007/s00037-016-0143-x.

25 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004.

26 Nathan Linial, Alex Samorodnitsky, and Avi Wigderson. A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents. *Combinatorica*, 20(4):545–568, 2000. doi:10.1007/s004930070007.

27 László Lovász. Singular spaces of matrices and their application in combinatorics. *Boletim da Sociedade Brasileira de Matemática-Bulletin/Brazilian Mathematical Society*, 20(1):87–99, 1989.

28 Vladimir L Popov. The constructive theory of invariants. *Izvestiya: Mathematics*, 19(2):359–376, 1982.

29 K.G. Ramanathan. *Lectures on the Algebraic Theory of Fields*. Tata Institute of Fundamental Research, Bombay, 1954.

30 Nitin Saxena. Progress on polynomial identity testing - II. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:186, 2013. URL: http://eccc.hpi-web.de/report/2013/186.

31 Aidan Schofield and Michel Van den Bergh. Semi-invariants of quivers for arbitrary dimension vectors. *Indagationes Mathematicae*, 12(1):125–138, 2001.

32 Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5:207–388, March 2010. doi:http://dx.doi.org/10.1561/0400000039.

33 Seinosuke Toda. Classes of arithmetic circuits capturing the complexity of computing the determinant. *IEICE Transactions on Information and Systems*, 75(1):116–124, 1992.

34 Kai-Tak Wong. The eigenvalue problem $\lambda Tx + Sx$. *Journal of Differential Equations*, 16(2):270 – 280, 1974. doi:10.1016/0022-0396(74)90014-X.

A Constructivizing the result of Derksen and Makam

Here is an algorithmic version of Lemma 2.7 of [12]. Let $M(k \times \ell, \mathbb{F})$ be the space of $k \times \ell$ matrices over $\mathbb{F}$, and define $\mathcal{B}^{\{k,\ell\}} := \mathcal{B} \otimes M(k \times \ell, \mathbb{F})$.

▶ **Lemma 18.** *Let $\mathcal{B} \leq M(n, \mathbb{F})$. Assume that for $k, \ell = 1, \ldots, N$ we are given matrices $M_0(k, \ell) \in \mathcal{B}^{\{k,\ell\}}$ of rank $r_0(k, \ell)$, and suppose that $|\mathbb{F}| \geq 2nN + 1$. Then for every $k, \ell = 0, \ldots, N$ we can efficiently (that is, by an algorithm that uses $\mathrm{poly}(Nn)$ arithmetic operations and, over e.g. $\mathbb{Q}$, produces intermediate and final data of size polynomial in the input size) construct matrices $M(k, \ell) \in \mathcal{B}^{\{k,\ell\}}$ of rank $r(k, \ell) \geq r_0(k, \ell)$ such that*

(1) $r(k, \ell + 1) \geq r(k, \ell)$ $(0 \leq \ell < N)$;
(2) $r(k + 1, \ell) \geq r(k, \ell)$ $(0 \leq k < N)$;
(3) $r(k, \ell + 1) \geq \frac{1}{2}(r(k, \ell) + r(k, \ell + 2))$ $(0 \leq \ell < N - 1)$;
(4) $r(k + 1, \ell) \geq \frac{1}{2}(r(k, \ell) + r(k + 2, \ell))$ $(0 \leq k < N - 1)$;
(5) $r(k, k)$ is divisible by k.

For $k = 0$ (resp. $\ell = 0$) we assume that $M_0(k, \ell)$ is the empty matrix having ℓ columns (resp. k rows), and $r(k, \ell) = 0$.

Proof. Initially put $M(k, \ell) = M_0(k, \ell)$ for every pair (k, ℓ). For a $k \times \ell$ matrix T let T^+ denote the $(k + 1) \times \ell$ matrix obtained form T by appending a zero ($(k + 1)$st) row, T^{++} is obtained by appending two zero rows.

For $M = \sum_{i=1}^{m} B_i \otimes T_i$ we use M^+ for $\sum_{i=1}^{m} B_i \otimes T_i^+$, while $M^{++} = \sum_{i=1}^{m} B_i \otimes T_i^{++}$.

Let (k, ℓ) be a pair such that any of (1)–(5) is violated. Then we will replace some of the matrices $M(k', \ell')$ with matrices having larger rank. Over an infinite base field like $\mathbb{Q}$, each such replacement step (or each small group consisting of a few them) can be followed by an application of the data reduction procedure from [10] to keep intermediate (as well as the final) data small.

If (1) is violated then, like in [12], replace $M(k + 1, \ell)$ with $M(k, \ell)^+$. We can treat a violation of (2) symmetrically.

When (3) is violated we consider the matrix $A = A(t) = M(k + 2, \ell) + tM(k, \ell)^{++}$ as a $(k + 2) \times \ell$ block matrix consisting of square blocks of size n from $\mathcal{B}$. We can choose t from any subset S of size $2nN + 1$ of the base field so that A has rank at least $r(k + 2, \ell)$, while the first kn rows form a matrix of rank at least $r(k, \ell)$. This is because a necessary condition for violating either of these two conditions is that the determinant of an appropriate (but unknown) sub-matrix vanishes which determinant is, as a polynomial of degree at most nN in t is not identically zero. The product of these polynomials has degree at most $2nN$ therefore it cannot have more that $2nN$ zeros.

If A has rank larger than $r(k+2, \ell)$ then we replace $M(k+2, \ell)$ with A. Otherwise, like in [12], let U be the span of the first kn rows of A, V be the span of the first $(k+1)n$ rows and W be the span of the first kn rows and the last n rows. Note that these collections rows correspond to matrices of the form $A_0 = \sum B_i \otimes T_i$, $A_1 = \sum B_i \otimes T_i'$ and $A_2 = \sum B_i \otimes T_i''$ where T_i are $k \times \ell$ matrices, while T_i' and T_i'' have $(k+1)$ rows and ℓ columns. As $U \leq V \cap W$ and the row space of A is $V + W$, we have $r(k, \ell) \leq \dim U \leq \dim(V \cap W) = \dim V + \dim W - \dim V + W = \dim V + \dim W - r(k + 2, \ell)$. It follows that $\dim V + \dim W \geq r(k, \ell) + r(k + 2, \ell)$, whence violation of (3) is only possible if either $\dim V$ or $\dim W$ is strictly larger than $\frac{1}{2}(r(k, \ell) + r(k + 2, \ell))$. Then we replace $M(k + 1, \ell)$ with A_1 or A_2, according to which one has larger rank. A violation of (4) is treated symmetrically.

When (5) is violated then we can apply Lemma 11.

As in each round when violation of (1),. . .,(4) or (5) occurs the rank of at least one of the matrices $M(k,\ell)$ is incremented, the total number of rounds for achieving (1)–(5) is at most $N^3 n$. ◀

And here is essentially Proposition 2.10 of [12]. We include a proof (which is almost literally the same as the proof in [12]) here for completeness. We note that this lemma deals only with the property of certain families of functions, without referring to matrices.

▶ **Lemma 19** ([12, Proposition 2.10]). *Assume that $N > n > 0$, $r : \{0,1,\ldots,N\}^2 \to \mathbb{Z}$ is a function with $0 \le r(k,\ell) \le \min(k,\ell)n$ for $k,\ell \in \{0,1,\ldots,N\}$ also satisfying (1)–(5) of Lemma 18. Suppose further that $r(1,1) > 1$, and there exists d such that $n \le d+1 \le N$ and $r(d+1,d+1) = n(d+1)$. Then, $r(d,d) = nd$ as well.*

Proof. By $r(d+1,d+1) = n(d+1)$, for $1 \le a < d+1$,

$$r(d+1,a) \ge \frac{(d+1-1)\cdot r(d+1,0) + a \cdot r(d+1,d+1)}{d+1} = an.$$

As by assumption $r(d+1,a) \le an$, we have $r(d+1,a) = an$. Similarly $r(a,d+1) = an$ for $1 \le a < d+1$.

Then we bound $r(1,d)$ as follows:

$$\begin{aligned} r(1,d) &\ge \frac{(d-1)\cdot r(1,d+1) + 1 \cdot r(1,1)}{d} \\ &\ge \frac{(d-1)n+2}{d} = n - \frac{n-2}{d} > n-1. \end{aligned}$$

Note that we use $r(1,1) > 1$ and $d \ge n-1$. Since $r(1,d) \in \mathbb{Z}$, $r(1,d) = n$.

We are ready to bound $r(d,d)$ then.

$$\begin{aligned} r(d,d) &\ge \frac{(d-1)\cdot r(d+1,d) + 1 \cdot r(1,d)}{d} \\ &= \frac{(d-1)dn+n}{d} = nd - n + \frac{n}{d}. \end{aligned}$$

From $d \ge n-1$ it is inferred easily that $-n + \frac{n}{d} > -d$. Therefore $nd - n + \frac{n}{d} > (n-1)d$. By (5) we conclude that $r(d,d) = nd$. ◀

We finally remark that, if we use Lemma 18 in the proof of Theorem 7, then n in the statement of the lemma will be $r+1$, N will be d', $M_0(d',d')$ is the nonsingular $(r+1)d' \times (r+1)d'$ block of A and $M_0(p,q)$ can be actually even the zero matrix for $(p,q) \ne (d',d')$. It will prepare matrices in several not necessarily square blow-ups, among others, most importantly, one in an (r,r)-blow-up with a similar content as described in the proof of Theorem 7.

B Efficient construction of cyclic field extensions of arbitrary degrees

A cyclic extension of a field $\mathbb{K}$ is a finite Galois extension of $\mathbb{K}$ having a cyclic Galois group. By constructing a cyclic extension $\mathbb{L}$ we mean constructing the extension as an algebra over $\mathbb{K}$, e.g., by giving an array of *structure constants* with respect to a $\mathbb{K}$-basis for $\mathbb{L}$ defining the multiplication on $\mathbb{L}$ as well as specifying a generator of the Galois group, e.g, by its matrix with respect to a $\mathbb{K}$-basis.

▶ **Lemma 20.** *Given a prime p and an integer $s \geq 1$, one can construct in time* $\mathrm{poly}(p^s)$ *a cyclic extension K_s of $\mathbb{F}_p(Z)$ of degree p^s such that $\mathbb{F}_p$ is algebraically closed in K_s. The field K_s will be given in terms of structure constants with respect to a basis over $\mathbb{F}_p(Z)$, and the generator σ for the Galois group will be given by its matrix in terms of the same basis. The structure constants as well as the entries of the matrix for σ will be polynomials in $\mathbb{F}_p[Z]$ of degree* $\mathrm{poly}(p^s)$.

Proof. First we briefly recall the general construction given in Section 6.4 of [29]. This, starting from a field K_0 of characteristic p, recursively builds a tower $K_0 < K_1 < \ldots < K_s$ of fields such that K_j is a cyclic extension of K_0 of degree p^j. Assume that K_s together with a K_0-automorphism σ_s of order p^s has already been constructed. (Initially let σ_0 be the identity map on K_0.) Then for any element $\beta_s \in K_s$ with $\mathrm{Tr}_{K_s:K_0}(\beta_s) = 1$ and for any $\alpha_s \in K_s$ such that $\alpha_s^{\sigma_s} - \alpha_s = \beta_s^p - \beta_s$ the polynomial $X^p - X - \alpha_s$ is irreducible in $K_s[X]$. (Existence of α_s with the required property follows from the additive Hilbert 90.) Put $K_{s+1} = K_s[X]/(X^p - X - \alpha_s)$ and let $\omega_{s+1} \in K_{s+1}$ be the image of of X under the projection $K_s[X] \to K_{s+1}$. Then σ_s extends to a K_0-automorphism σ_{s+1} of degree p^{s+1} of K_{s+1} such that $\omega_{s+1}^{\sigma_{s+1}} = \omega_{s+1} + \beta_s$. This gives a cyclic extension of degree p^{s+1}.

Now we specify some details of a polynomial time construction for $K_0 = \mathbb{F}_p(Z)$. In the first step we take $\beta_0 = 1$, and, in order to guarantee that the only elements in K_1 which are algebraic over $\mathbb{F}_p$ is F_p (we also use the phrase F_p is algebraically closed in K_1 when this property holds), we take $\alpha_0 = Z$. Then K_1 is a pure transcendental extension of $\mathbb{F}_p$. As K_s/K_0 is a cyclic extension of oder p^s, it has a unique subfield which is an order p extension of K_0. This must be K_1. Then $\mathbb{F}_p$ has no proper finite extension in K_s as otherwise K_0 would also have another degree p extension.

We consider the following K_0-basis for K_s:

$$\sigma_s = \left\{ \prod_{j=1}^{s} \omega_j^k, \quad (k = 0, \ldots, p-1) \right\},$$

where ω_j is a root of $X^p - X - \alpha_{j-1}$ in K_j. We claim that $\mathrm{Tr}_{K_j:K_{j-1}}(\omega_j^{p-1}) = -1$. Indeed, in the K_{j-1}-basis $\omega_j^0, \ldots, \omega_j^{p-1}$ for K_j, in the matrix of multiplication by ω_j^{p-1} the diagonal entries consist of $p-1$ ones and one zero. Therefore $\mathrm{Tr}_{K_j:K_{j-1}}(\omega_j^{p-1}\sigma) = -\sigma$ for every $\sigma \in K_{j-1}$, whence $\mathrm{Tr}_{K_j:K_0}(\omega_j^{p-1}\sigma) = -\mathrm{Tr}_{K_{j-1}:K_0}(\sigma)$. Now by induction we obtain $\mathrm{Tr}_{K_j:K_0} \prod_{i=1}^{j} \omega_i^{p-1} = (-1)^j$. Therefore in each step (when $j > 0$) we can choose $\beta_j = (-1)^j \prod_{i=1}^{j} \omega_i^{p-1}$ and α_j thereafter, following the construction in the standard proof of the additive Hilbert 90. Specifically, we set

$$\alpha_j = (-1)^{j+1} \sum_{k=1}^{p^j-1} \beta_j^{\sigma_j^k} \left(\sum_{\ell=0}^{k-1} (\beta_j^p - \beta_j)^{\sigma_j^\ell} \right). \tag{B.1}$$

Then $\alpha_j^{\sigma_j} - \alpha_j = \beta_j^p - \beta_j$. Notice that α_j is a sum of terms with each of which, up to a sign, is a product of at most $p+1$ conjugates $\beta_j^{\sigma_j^\ell}$ (with various ℓs) of β_j ($\ell \leq p^j$)

Assume by induction that the structure constants of K_j with respect to the basis σ_j are polynomials from $\mathbb{F}_p[Z]$ of degree at most Δ_j and the same holds for the entries of the matrix of σ_j^ℓ for every $1 \leq \ell < p^j$ (written in the same basis). For $j = 1$ this holds with $\Delta_1 = 1$. (To see this, observe that for $0 \leq k, \ell < p$, the product $\omega_1^k \omega_1^\ell$ is the basis element of $\omega_1^{k+\ell}$ if $k + \ell < p$, while otherwise it equals the sum $\omega_1^{k+\ell-p+1} + Z\omega_1^{k+\ell-p}$.) Then, if we express α_j in terms of the basis σ_j using Eq. B.1, we obtain that its coordinates are polynomials of

degree at most $(2p+1)\Delta_j$. This is because $(-1)^j\beta_j \in \sigma_j$, whence $\beta_j^{\sigma^\ell}$ has coordinates of polynomials of degree bounded by Δ_j. In Eq. B.1, we have the products of at most $p+1$ such elements, so the result will have polynomial coordinates of degree at most $(2p+1)\Delta_j$.

Now consider the product of two elements $\omega_{j+1}^k\sigma_1$ and $\omega_{j+1}^\ell\sigma_2$ of σ_{j+1}. Here $k,\ell < p$ and $\sigma_1,\sigma_2 \in \sigma_j$. The coordinates of the product $\sigma_1\sigma_2$ with respect to σ_j are polynomials of degree at most Δ_j. The same holds for the product $\omega_{j+1}^{k+\ell}\sigma_1\sigma_2$ if $k+\ell < p$. If $k+\ell > p$, then $\omega_{j+1}^{k+\ell} = \omega_{j+1}^p\omega_{j+1}^{k+\ell-p} = (\omega_{j+1}+\alpha_j)\omega_{j+1}^{k+\ell-p}$, whence $\omega_{j+1}^{k+\ell}\sigma_1\sigma_2$ is the sum of $\omega_{j+1}^{1+k+\ell-p}\sigma_1\sigma_2$ and $\alpha_j\sigma_1\sigma_2$. The former term has coordinates of degree at most Δ_j, the coordinates of the latter are polynomials of degree at most $(2p+1)\Delta_j + \Delta_j + \Delta_j = (2p+3)\Delta_j$.

Now consider the conjugate of $\omega_{j+1}^k\sigma$ by σ_{j+1}^ℓ, where $1 \le \ell < p^{j+1}$, $1 \le k \le p-1$ and $\sigma \in \sigma_j$. This conjugate is $(\omega_{j+1}^{\sigma_{j+1}^\ell})^k\sigma^{\sigma_{j+1}^\ell}$. The second term equals $\sigma^{\sigma_j^\ell}$ which has coordinates of degree at most Δ_j. To investigate the first term, recall that $\omega_{j+1}^{\sigma_{j+1}} = \omega_{j+1} + \beta_j$, whence

$$\omega_{j+1}^{\sigma_{j+1}^\ell} = \omega_{j+1} + \sum_{r=0}^{\ell-1}\beta_j^{\sigma_j^r}$$

The element $\delta = \sum_{r=0}^{\ell-1}\beta_j^{\sigma_j^r}$, expressed in terms of σ_j, has again polynomial coordinates of degree at most Δ_j. Then $(\omega_{j+1}^{\sigma_{j+1}^\ell})^k$ is the sum (with binomial coefficients) of terms of the form $\omega_{j+1}^r\delta^{k-r}$. The power δ^{k-r} has coordinates of degree at most $(k-r)\Delta_j + (k-r-1)\Delta_j \le (2p-1)\Delta_j$ in terms of σ_j, whence we conclude that $(\omega_{j+1}^{\sigma_{j+1}^\ell})^k$ has, in terms of σ_{j+1} polynomial coordinates of degree at most $(2p-1)\Delta_j$. It follows that the matrix of any power of σ_{j+1} has polynomial entries of degree at most $2p\Delta_j$.

We obtained that the function $(2p+3)^s = \mathrm{poly}(p^s)$ is an upper bound for both the structure constants and for the matrices of the powers of σ_s. ◀

▶ **Lemma 17** (restated). *Let $\mathbb{F}'$ be a field. Let d be any non-negative integer. If $\mathrm{char}(\mathbb{F}') = 0$ then $d_1 = d$. If $\mathrm{char}(\mathbb{F}') = p > 0$ then let d_1 be the p-free part of d, that is, $d = d_1p^s$, where $p \nmid d_1$ and $s \in \mathbb{N}$. Assume that $\mathbb{F}'$ contains a known d_1th root of unity ζ. Then a cyclic extension $\mathbb{L}$ degree d of $\mathbb{K} := \mathbb{F}'(X)$ can be computed using $\mathrm{poly}(d)$ arithmetic operations. $\mathbb{L}$ will be given by structure constants with respect to a basis, and the matrix for a generator of the Galois group in terms of the same basis will also be given. All the output entries (the structure constants as well as the entries of the matrix representing the Galois group generator) will be polynomials of degree $\mathrm{poly}(d)$ in $\mathbb{F}'[X]$. Furthermore for $\mathbb{F}' = \mathbb{Q}[\sqrt[d_1]{1}]$, the bit complexity of the algorithm (as well as the size of the output) is $\mathrm{poly}(d)$.*

Proof. Put $\mathbb{L}_1 = \mathbb{F}'(Y)$ and $X = Y_1^{d_1}$. Then $1, Y_1, \ldots, Y_1^{d_1}$ are a $\mathbb{F}'(X)$-basis for $\mathbb{L}_1$ with $Y_1^iY_1^j = Y_1^{i+j}$ if $i+j \le d_1$ and $XY_1^{i+j-d_1}$ otherwise. Further note that the linear extension σ_1 of the map sending Y_1^j to $\zeta^jY_1^j$ is an automorphism of degree d_1. Then $\mathbb{L}_1$ is a cyclic extension of $\mathbb{F}'(X)$ of degree d_1. This procedure has been used in [24].

We can compute whether $\mathrm{char}(\mathbb{F}')$ is a divisor of d by testing the multiples of the identity element up to d. If $\mathrm{char}(\mathbb{F}') = 0$, or if $\mathrm{char}(\mathbb{F}') = p > 0$ and $p \nmid d$, we are done. Note that in the following $p \le d$.

If $\mathrm{char}(\mathbb{F}') = p > 0$ and $p \mid d$, let d_1 be in the statement, so $d = d_1p^s$. Let $d_2 = p^s$, and $\mathbb{F}_p$ be the prime field of $\mathbb{F}'$. Construct the cyclic extension of degree d_2 of $\mathbb{F}_p(X)$ over $\mathbb{F}_p$ by Lemma 20, and let the resulting field be $\mathbb{L}_2$. We also obtain the matrix a generator σ_2 of the Galois group. Then put $\mathbb{L} = \mathbb{L}_1 \otimes_{\mathbb{F}_p(X)} \mathbb{L}_2$. It contains a copy of $\mathbb{K} = \mathbb{F}'(X) \cong \mathbb{F}'(X) \otimes_{\mathbb{F}_p(X)} \mathbb{F}_p(X)$. We take the product basis for the structure constants and for matrix representation of the automorphism $\sigma_1 \otimes \sigma_2$. ◀

The Duality Gap for Two-Team Zero-Sum Games

Leonard J. Schulman[*1] **and Umesh V. Vazirani**[2]

1 **Caltech, Engineering & Applied Science MC305-16, Pasadena, USA**
schulman@caltech.edu

2 **UC Berkeley, Computer Science, Berkeley, USA**
vazirani@eecs.berkeley.edu

Abstract

We consider multiplayer games in which the players fall in two teams of size k, with payoffs equal within, and of opposite sign across, the two teams. In the classical case of $k = 1$, such zero-sum games possess a unique value, independent of order of play, due to the von Neumann minimax theorem. However, this fails for all $k > 1$; we can measure this failure by a duality gap, which quantifies the benefit of being the team to commit last to its strategy. In our main result we show that the gap equals $2(1 - 2^{1-k})$ for $m = 2$ and $2(1 - m^{-(1-o(1))k})$ for $m > 2$, with m being the size of the action space of each player. At a finer level, the cost to a team of individual players acting independently while the opposition employs joint randomness is $1 - 2^{1-k}$ for $k = 2$, and $1 - m^{-(1-o(1))k}$ for $m > 2$.

This class of multiplayer games, apart from being a natural bridge between two-player zero-sum games and general multiplayer games, is motivated from Biology (the weak selection model of evolution) and Economics (players with shared utility but poor coordination).

1998 ACM Subject Classification G.1.6 Optimization

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.56

1 Introduction

Games between *teams* of players are ubiquitous; in the economy this occurs most prominently in competition between firms. Another case of significance is that in which a team is a biological species and the players on the team are the genes of the species. What makes a set of players a team, in our idealization, is that in any outcome the players in the set receive identical payoffs.

Competition among firms or species need not be zero-sum; however, the zero-sum case will be the focus of this paper, being the most basic form of competition, and often an approximation to reality. Specifically, a two-team zero-sum game is a multiplayer game in which the players are partitioned into two sets A and B, and a real-valued payoff tensor (of dimension equal to the number of players) specifies the payoff conditional on player actions; this payoff accrues positively to each player of Team B and negatively to each player of Team A.

If perfect coordination within each team can be achieved, then a zero-sum interaction between two teams is nothing but a zero-sum interaction between two "virtual" players. In the biological setting, the opposite extreme is relevant: an important model in evolutionary theory is the *weak selection* model (see [8, 1, 10, 9, 13, 14, 3, 7]), in which a species is a team, the genes are the players, the alleles of the gene are the possible actions of a player, and the

* LJS was supported in part by NSF grants 1319745/1618795 and BSF grant 2012333; UVV was supported in part by NSF Grant CCF-1410022.


© Leonard Schulman and Umesh Vazirani;
licensed under Creative Commons License CC-BY

allele frequencies are independent across genes. Likewise, the difficulty of coordination has long been treated in the economic literature as one of the forces limiting the size of firms [4].

This raises a natural question: does von Neumann's minimax theorem for zero-sum games continue to hold for a zero-sum team game where the individual members of the team play independently, and if not can the deviation be bounded? Formally, this deviation is expressed through the *duality gap* between the values of the game (always expressed as the payoff to Team B) under two scenarios: when all members of Team A must first commit to their randomized strategies, and then Team B gets to respond; and when all members of Team B must commit, and then Team A gets to respond. Surprisingly, this natural question does not seem to have been asked before.

In Section 3 we determine the range of this duality gap: for action spaces of size 2 we give the exact answer for all k (the size of the teams); for action spaces of any size $m > 2$ we determine the asymptotics of the answer as a function of k.

The key lemma in the lower bound on the duality gap for $m > 2$ may be of independent interest: fix a random set S of $g(m)$ m-ary strings of any length k. Then with high probability, any product distribution may place probability more than $m^{-k(1-o(1))}$ (i.e. much more than the uniform distribution would) on at most $O(\log g(m))$ strings in S.

Another interesting quantity is the *defensive gap*: the defensive gap of Team B is the difference between the payoff to Team B if Team A must play a product strategy, and the payoff to Team B if Team A can use a joint (but hidden from B) source of randomness. The defensive gap quantifies the reduced effectiveness of a team when its members are constrained to choose their actions from a product distribution. Note that the duality gap is the sum of the two teams' defensive gaps.

The defensive gap may be compared to two notions in the existing literature. Assume Team A goes first, and think just of the multiplayer game being played by the k players of Team A. (Since Team B can respond optimally, even deterministically, once the strategies of Team A have been fixed, we may ignore the players of Team B and consider their response merely part of the definition of the game being played by Team A.) Then, since the payoffs to all players in Team A are identical, the defensive gap is the difference between the value of the best correlated equilibrium [2] and the best Nash equilibrium [12]. It quantifies, if you will, the penalty for not coordinating. Again, since the payoffs are identical within the team, social welfare agrees with individual welfare, and so the defensive gap is, conceptually, a Price of Stability [11] (although that "price" is normally defined as a ratio and not a difference).

Finally, in Section 4 we go on to investigate how the value of a game can be affected by more incremental changes, specifically, by exchanging the order of play of two players of opposing teams who were playing (committing to their randomized strategy) in immediate succession. There are games with duality gap bounded away from 0, in which all these value changes tend to 0 in k; whereas there are other games, including games symmetric within each team, in which the largest such value change is bounded away from 0 as a function of k.

2 Preliminaries

We consider two-team zero-sum games in which Team A has k players each with m choices in its action space; likewise for Team B. The payoff (to Team B) is specified by a tensor T in $(\mathbb{R}^m)^{\otimes 2k}$. (In the biological setting, each of A and B is a species with a genome of k genes, each taking on one of m possible alleles.) More formally to each player A_i, $i = 1, \ldots k$ of Team A corresponds a vector space $U_i \cong \mathbb{R}^m$, and to each player B_j, $j = 1, \ldots k$ of Team

B, a vector space $V_j \cong \mathbb{R}^m$. Space U_i is spanned by a basis which we denote $u_{i0}, \ldots, u_{i,m-1}$. Similarly for V_j. In this setting $T \in U_1^* \otimes \ldots \otimes U_k^* \otimes V_1^* \otimes \ldots \otimes V_k^*$. (With $*$ denoting dualization.) Letting $\boldsymbol{I} = (i_1, \ldots, i_k) \in \{0, \ldots, m-1\}^k$ represent an action of the players of Team A, and $\boldsymbol{J} = (j_1, \ldots, j_k) \in \{0, \ldots, m-1\}^k$ an action of the players of Team B, $T_{\boldsymbol{I}}^{\boldsymbol{J}}$ is the payoff to players of Team B (and minus the payoff to players of Team A).

In the case $m = 2$, the strategy of player A_i is specified by a parameter $0 \le p_i \le 1$ which is the probability with which he plays choice 0, i.e., vector u_{i0}. Likewise player B_j has a parameter $0 \le q_j \le 1$ which is the probability with which he plays choice 0, i.e., vector v_{j0}. For $m > 2$, the strategy of player A_i is specified by a probability distribution $(p_{i0}, \ldots, p_{i,m-1})$ and the strategy of player B_j is specified by a probability distribution $(q_{j0}, \ldots, q_{j,m-1})$. (Thus for $m = 2$, $(p_i, 1-p_i)$ is shorthand for (p_{i0}, p_{i1}).)

We let T_p^q denote the expected payoff to Team B when the players of Team A use distributions p_i and those of Team B use distributions q_j. This notation generalizes the notation $T_{\boldsymbol{I}}^{\boldsymbol{J}}$, if one interprets $\boldsymbol{I}$ as the probability distribution on $\{0, \ldots, m-1\}^k$ supported solely on $\boldsymbol{I}$ (and similarly for $\boldsymbol{J}$). Equivalently, T_p^q equals the scalar given by contracting T with the tensor product of the vectors $(p_{i0}, \ldots, p_{i,m-1})$ (ranging over i) and $(q_{j0}, \ldots, q_{j,m-1})$ (ranging over j).

By a standard argument, $\min_p \max_q T_p^q \ge \max_q \min_p T_p^q$. (We write everywhere min or max rather than inf or sup since the spaces are compact.) However, apart from the linear ($k = 1$) case, the gap $\min_p \max_q T_p^q - \max_q \min_p T_p^q$ can be positive.

Our purpose is to quantify this gap relative to the uniform norm $\|T\|_\infty = \max_{\boldsymbol{I},\boldsymbol{J}} |T_{\boldsymbol{I}}^{\boldsymbol{J}}|$. We define the duality gap of tensor T:

$$\mathrm{gap}(T) = \frac{\min_p \max_q T_p^q - \max_q \min_p T_p^q}{\|T\|_\infty} = \frac{\min_p \max_{\boldsymbol{J}} T_p^{\boldsymbol{J}} - \max_q \min_{\boldsymbol{I}} T_{\boldsymbol{I}}^q}{\|T\|_\infty} \tag{2.1}$$

where as above, $\boldsymbol{I}$ or $\boldsymbol{J}$ represent the pure strategy choosing that action.

The principal quantity of interest is

$$\mathrm{gap}_{m,k} = \max_T \mathrm{gap}(T) \tag{2.2}$$

ranging over games T for teams of size k and action spaces of size m. It is trivial that $\mathrm{gap}_{m,k} \le 2$. Moreover $\mathrm{gap}_{m,k}$ is nondecreasing in k (because one may ignore the actions of players after the k'th player on each team), and in m (because one may map all actions $\ge m-1$ to action $m-1$).

Here and throughout the paper, upper-case P and Q denote mixed strategies of virtual players; that is to say, each is a general probability tensor (a tensor with nonnegative entries summing to 1), $P \in U_1 \otimes \ldots \otimes U_k$ and $Q \in V_1 \otimes \ldots \otimes V_k$. Lower-case p and q denote product distributions, i.e., rank-one probability tensors.

Extending the existing notation, T_P^Q is the expected payoff to B when A (as a virtual player) uses distribution P and B uses distribution Q. It is also useful to employ the standard convention that a repeated index indicates tensor contraction over that index, so $P^{\boldsymbol{I}} T_{\boldsymbol{I}}^{\boldsymbol{J}} = T_P^{\boldsymbol{J}} \in V_1^* \otimes \ldots \otimes V_k^*$ and $Q_{\boldsymbol{J}} T_{\boldsymbol{I}}^{\boldsymbol{J}} = T_{\boldsymbol{I}}^Q \in U_1^* \otimes \ldots \otimes U_k^*$.

By strong LP duality we can define the value of the virtual player game by

$$\mathrm{Val}\, T = \min_P \max_{\boldsymbol{J}} \{P^{\boldsymbol{I}} T_{\boldsymbol{I}}^{\boldsymbol{J}}\} = \max_Q \min_{\boldsymbol{I}} \{Q_{\boldsymbol{J}} T_{\boldsymbol{I}}^{\boldsymbol{J}}\}. \tag{2.3}$$

Let P and Q be strategies achieving equality in (2.3). We can usefully refine the study of $\mathrm{gap}(T)$ by defining the *defensive gap* of Team A in tensor T as

$$\mathrm{gap}_A(T) = \frac{\min_p \max_{\boldsymbol{J}} \{p^{\boldsymbol{I}} T_{\boldsymbol{I}}^{\boldsymbol{J}}\} - \max_{\boldsymbol{J}} \{P^{\boldsymbol{I}} T_{\boldsymbol{I}}^{\boldsymbol{J}}\}}{\|T\|_\infty} = \frac{\min_p \max_{\boldsymbol{J}} \{p^{\boldsymbol{I}} T_{\boldsymbol{I}}^{\boldsymbol{J}}\} - \mathrm{Val}\, T}{\|T\|_\infty}$$

where, of course, p ranges over product distributions. Likewise the defensive gap of Team B is

$$\mathrm{gap}_B(T) = \frac{\min_{\boldsymbol{I}}\{Q_{\boldsymbol{J}} T^{\boldsymbol{J}}_{\boldsymbol{I}}\} - \max_q \min_{\boldsymbol{I}}\{q_{\boldsymbol{J}} T^{\boldsymbol{J}}_{\boldsymbol{I}}\}}{\|T\|_\infty} = \frac{\mathrm{Val}\, T - \max_q \min_{\boldsymbol{I}}\{q_{\boldsymbol{J}} T^{\boldsymbol{J}}_{\boldsymbol{I}}\}}{\|T\|_\infty}$$

The defensive gap quantifies the reduced effectiveness of a team of players (when forced to commit to a mixed strategy to which the other team has a chance to respond), as compared with a virtual player (in the same situation).

3 The Defensive Gaps and the Duality Gap

▶ **Theorem 1.** $\mathrm{gap}_{2,k} = 2(1-2^{1-k})$, *and for every* $m > 2$, $2(1-m^{-(1-o(1))k}) \le \mathrm{gap}_{m,k} \le 2(1-m^{1-k})$. *(The "*$o(1)$*" being w.r.t.* k*.) More specifically:*

1. *Upper bound on the defensive gaps and duality gap:*
   For any $m \ge 2$, *and for any* T *having action spaces of size* m, $\mathrm{gap}_A(T) \le (1 - \mathrm{Val}\, T/\|T\|_\infty)(1-m^{1-k})$ *and* $\mathrm{gap}_B(T) \le (1+\mathrm{Val}\, T/\|T\|_\infty)(1-m^{1-k})$.
2. *Lower bound on the duality gap:*
   (a) $\mathrm{gap}_{2,k} \ge 2(1-2^{1-k})$.
   (b) *For every* $m > 2$ *there is a function* $\varepsilon(k)$ *tending to* 0 *as* $k \to \infty$, *such that* $\mathrm{gap}_{m,k} \ge 2(1-m^{-(1-\varepsilon(k))k})$.

Henceforth scale any $T \neq 0$ so that $\|T\|_\infty = 1$.

Proof.

Part (1): Upper Bounds on $\mathrm{gap}_A(T)$, $\mathrm{gap}_B(T)$ and $\mathrm{gap}_{m,k}$.

It suffices to show the claim for gap_A. The claim for gap_B follows by negating all entries of T, reversing the roles of the teams and applying the claim for gap_A. The claim for the duality gap follows because $\mathrm{gap}(T) = \mathrm{gap}_A(T) + \mathrm{gap}_B(T)$.

We start with the case $m = 2$. Given an arbitrary coordinated mixed strategy P for team A, we wish to "round" it to a rank-one probability tensor (a product distribution) that does no worse than the claimed defensive gap. The natural candidate for rounding would be the approximation by independent random variables having the same marginals as P. That is, set

$$p_1 = \sum_{i_2,\ldots,i_k} P^{0,i_2,\ldots,i_k}, \quad p_2 = \sum_{i_1,i_3,\ldots,i_k} P^{i_1,0,i_3,\ldots,i_k}, \quad \text{etc.} \tag{3.1}$$

and, letting

$$p = (p_1, 1-p_1) \otimes \ldots \otimes (p_k, 1-p_k), \tag{3.2}$$

use p as the rank-one strategy replacing P, and use it to bound the defensive gap. It turns out that this approach cannot be used to prove any bound on the defensive gap. Surprisingly, there is a less obvious rank-one strategy which can be obtained from P and which provides a tight bound on the defensive gap.

For $0 \le x \le 1/2$ set

$$\beta(x) = \frac{(2x)^{1/k}}{2} \tag{3.3}$$

and for $1/2 < x \le 1$ set $\beta(x) = 1 - \beta(1-x)$. (Observe that β is increasing.)

Then use the rank-one strategy:

$$\tilde{p} = (\beta(p_1), \beta(1-p_1)) \otimes \ldots \otimes (\beta(p_k), \beta(1-p_k)) \tag{3.4}$$

We now claim that for every $\boldsymbol{I}$, $\tilde{p}^{\boldsymbol{I}} \geq 2^{1-k} P^{\boldsymbol{I}}$. We show this for $\boldsymbol{I} = \mathbf{0}$; all other $\boldsymbol{I}$ follow by the same argument. First note that for every $1 \leq i \leq k$, $p_i \geq P^{\mathbf{0}}$, and therefore $\beta(p_i) \geq \beta(P^{\mathbf{0}})$. So $\tilde{p}^{\mathbf{0}} = \beta(p_1) \cdots \beta(p_k) \geq \beta(P^{\mathbf{0}})^k$. Now there are two cases. If $P^{\mathbf{0}} \leq 1/2$ then $\beta(P^{\mathbf{0}})^k = 2^{1-k} P^{\mathbf{0}}$ as desired. If $P^{\mathbf{0}} > 1/2$ then $\beta(P^{\mathbf{0}})^k = (1 - \frac{1}{2}(2 - 2P^{\mathbf{0}})^{1/k})^k$. Showing this is $\geq 2^{1-k} P^{\mathbf{0}}$ is equivalent to showing that

$$(2 - (2P^{\mathbf{0}})^{1/k})^k \geq 2 - 2P^{\mathbf{0}}. \tag{3.5}$$

Let $\varepsilon = (2P^{\mathbf{0}})^{1/k} - 1$ and note $0 \leq \varepsilon \leq 1$. Then (3.5) is equivalent to $(1-\varepsilon)^k + (1+\varepsilon)^k \geq 2$ which holds by the power-mean inequality (for k vs. 1).
From $\tilde{p}^{\boldsymbol{I}} \geq 2^{1-k} P^{\boldsymbol{I}}$ for every $\boldsymbol{I}$ and from $\|T\|_\infty \leq 1$, we have that for every $\boldsymbol{J}$:

$$\tilde{p}^{\boldsymbol{I}} T^{\boldsymbol{J}}_{\boldsymbol{I}} \leq 1 - 2^{1-k} + 2^{1-k} P^{\boldsymbol{I}} T^{\boldsymbol{J}}_{\boldsymbol{I}} \tag{3.6}$$

which (upper bounding $P^{\boldsymbol{I}} T^{\boldsymbol{J}}_{\boldsymbol{I}}$ by $\operatorname{Val} T$, and subtracting $\operatorname{Val} T$ from each side) completes the proof for $m = 2$.
The same proof technique works for $m > 2$, only there is more flexibility in the function β. Given that in an optimal virtual-player distribution P for Team A, the marginal distribution of player i is $p_i(0), \ldots, p_i(m-1)$, we require a new distribution $\beta_i(0), \ldots, \beta_i(m-1)$ for player i such that for every ℓ, $\beta_i(\ell)^k \geq m^{1-k} p_i(\ell)$, which is to say, $\beta_i(\ell) \geq (m p_i(\ell)^{1/k})/m$. Such a distribution exists due to the inequality $\sum_\ell (m p_i(\ell))^{1/k}/m \leq 1$, which holds because by the power-mean inequality $(\sum_\ell (m p_i(\ell))^{1/k}/m)^k \leq \sum_\ell (m p_i(\ell))/m = 1$. The rest of the details are as for $m = 2$.

Part (2a): Lower Bound on $\mathrm{gap}_{2,k}$.

Write $\mathbf{0} = (0, \ldots 0)$ and $\mathbf{1} = (1, \ldots 1)$.
Consider the following tensor.

▶ **Example 2.**

$$\begin{cases} G^{\mathbf{0}}_{\boldsymbol{I}} & = \begin{cases} -1 & \text{if } \boldsymbol{I} = \mathbf{0} \\ 1 & \text{otherwise} \end{cases} \\ G^{\mathbf{1}}_{\boldsymbol{I}} & = \begin{cases} -1 & \text{if } \boldsymbol{I} = \mathbf{1} \\ 1 & \text{otherwise} \end{cases} \\ G^{\boldsymbol{J}}_{\mathbf{0}} & = \begin{cases} 1 & \text{if } \boldsymbol{J} = \mathbf{1} \\ -1 & \text{otherwise} \end{cases} \\ G^{\boldsymbol{J}}_{\mathbf{1}} & = \begin{cases} 1 & \text{if } \boldsymbol{J} = \mathbf{0} \\ -1 & \text{otherwise} \end{cases} \\ G^{\boldsymbol{J}}_{\boldsymbol{I}} & = 0 \text{ for all other } \boldsymbol{I}, \boldsymbol{J} \end{cases} \tag{3.7}$$

An informal description of this game is that if both Team A and Team B choose actions in $\{\mathbf{0}, \mathbf{1}\}$, then the outcome is as it would be in the "matching pennies" game. If just one of the teams chooses an action in $\{\mathbf{0}, \mathbf{1}\}$, then that team wins. If neither team chooses an action in $\{\mathbf{0}, \mathbf{1}\}$, then the game is a tie.
(We incidentally note that the proof of Part (2a) does not depend on setting entries to 0 in the last line of 3.7; the argument will go through with each entry taking any value in $[-1, 1]$.)

Now consider any strategy $p = (p_1, \ldots, p_k)$ for Team A (recall these are the probabilities of action 0). The expected payoff for action $\boldsymbol{J} = \mathbf{0}$ of Team B is

$$G_p^{\mathbf{0}} = 1 - 2p_1 \cdots p_k$$

The expected payoff for $\boldsymbol{J} = \mathbf{1}$ is

$$G_p^{\mathbf{1}} = 1 - 2(1 - p_1) \cdots (1 - p_k)$$

By the arithmetic-geometric mean inequality, $G_p^{\mathbf{0}} \geq 1 - 2(\frac{1}{k}\sum p_i)^k$ and $G_p^{\mathbf{1}} \geq 1 - 2(\frac{1}{k}\sum(1-p_i))^k$. So

$$\frac{1 - \max\{G_p^{\mathbf{0}}, G_p^{\mathbf{1}}\}}{2} \leq \min\{(\frac{1}{k}\sum p_i)^k, (\frac{1}{k}\sum(1-p_i))^k\}$$

equivalently

$$\left(\frac{1 - \max\{G_p^{\mathbf{0}}, G_p^{\mathbf{1}}\}}{2}\right)^{1/k} \leq \min\{\frac{1}{k}\sum p_i, \frac{1}{k}\sum(1-p_i)\}$$

The RHS is at most $1/2$. So $\min_p \max\{G_p^{\mathbf{0}}, G_p^{\mathbf{1}}\} \geq 1 - 2^{1-k}$.
A similar argument applied to the strategy q of Team B establishes that
$\max_q \min\{G_{\mathbf{0}}^q, G_{\mathbf{1}}^q\} \leq -1 + 2^{1-k}$. Adding the two contributions, Part (2a) of the theorem follows.

Part (2b): Lower Bound on $\mathbf{gap}_{m,k}$, $m > 2$.

Proof: We non-constructively exhibit a game establishing the lower bound. We start by selecting, for a function $g(m)$ to be specified, $g(m)$ strings $S = (s^1, \ldots, s^{g(m)})$, each s^j chosen independently and uniformly in $\{0, \ldots, m-1\}^k$ (Think of m as arbitrary but fixed while $k \to \infty$). The first idea of the proof lies in the interesting fact that with high probability, this set (whose size is independent of k) has the property that any product distribution may place probability more than $m^{-k(1-o(1))}$ on at most $\sim \log g(m)$ strings in S.

To describe the second idea, we start by associating the strings of S with players in a tournament. (A tournament is a digraph in which there is one directed edge between every pair of distinct vertices.) The game is then as follows. Associate the strings of S with the vertices of the tournament in an arbitrary way. If neither team chooses a string in S, the game is a tie. If one team chooses a string in S and the other does not, the first team wins. If both teams choose strings in S, the winning vertex is that which points toward the other (with a tie for identical strings). In all cases a win means a payoff of 1 and a tie a payoff of 0.

Key to the second idea is that by a result of Erdös [5] (later constructively in [6]) it is possible to choose the tournament so that it has no dominating set of size $\sim \log g(m)$. (A set of vertices is dominating if every vertex outside the set is pointed to by at least one vertex in the set.)

Now we claim that whichever team goes second can achieve payoff $1 - m^{-(1-o(1))k}$. The argument is the same for both teams, so say Team A goes first and let p be its product strategy. Let R be the $\sim \log g(m)$ strings in S accorded highest probability in p; all other strings of S have probability less than $m^{-(1-o(1))k}$. Team B responds to p with an $s \in S$ which dominates all of R. Team B wins unless Team A selects an $s' \in S - R$ (and sometimes even then). The payoff to Team B is therefore at least $1 - g(m)m^{-(1-o(1))k}$ which is at least $1 - m^{-(1-o(1))k}$. ◀

4 Order Refinements

It is natural to consider a more general scenario in which players of the two teams commit to their strategies in some (not necessarily strict) alternation. That is to say, let π be any bijection from $\{1,\ldots,2k\}$ to $\{A_1,\ldots,A_k,B_1,\ldots,B_k\}$. If $\pi(\ell)=A_i$ for some i then let $M(\ell)$ be the quantifier $\min_{p_i}$; if $\pi(\ell)=B_j$ for some j then let $M(\ell)=\max_{q_j}$. Then the value of game T with respect to order π is defined to be

$$V(T,\pi)=M(1)\ldots M(2k)T_p^q.$$

In particular, let π^{AB} be an order in which all the members of Team A go first, that is, $\pi^{AB}(\ell)=A_\ell$ for $\ell\le k$, and $\pi^{AB}(\ell)=B_{\ell-k}$ for $\ell>k$. (Note that V is invariant under exchange of same-team players with adjacent quantifiers.) Likewise let π^{BA} be an order in which Team B goes first. Then the duality gap of game T is

$$\mathrm{gap}(T)=V(T,\pi^{AB})-V(T,\pi^{BA}).$$

We now ask how much V may change when we change π by a single adjacent transposition.

A natural place to study this question is in *symmetric games*, by which we mean games invariant under permutation of the actions taken by members of a team. Even in this restrictive setting, a wide range of behaviors can occur.

For one extreme, we return to the game G of Example 2. We show that for any k, the order of the first three players can affect the outcome decisively.

▶ **Lemma 3.** *If $\ell\ge 3$ is the first time that Team B plays in π, then $V(G,\pi)\ge 1/2$. Likewise if $\ell\ge 3$ is the first time that Team A plays in π, then $V(G,\pi)\le -1/2$.*

For the other extreme, we have:

▶ **Theorem 4.** *There exist symmetric team games with any $k\ge 1$ players per team, with duality gap bounded away from 0 (as a function of k), but in which any adjacent transposition in the order of play changes the value of the game only by $O(1/k)$.*

Proofs of the lemma and theorem will be provided in the journal version of the paper.

5 Discussion

We have characterized the possible range of the duality gap. The examples which achieved large gap were highly structured. It would be interesting to find natural conditions on a game (particularly a symmetric game) that ensure small duality gap.

It would be interesting to extend our inquiry to teams (possibly more than two) competing in non-zero-sum games.

References

1 H Akashi. Inferring weak selection from patterns of polymorphism and divergence at “silent" sites in Drosophila DNA. *Genetics*, 139(2).

2 R Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1:67–96, 1974.

3 E Chastain, A Livnat, C Papadimitriou, and U Vazirani. Algorithms, games, and evolution. *Proc. National Academy of Sciences*, 2014.

4 R H Coase. The nature of the firm. *Economica, New Series*, 4(16):386–405, Nov. 1937.

5 P Erdös. On a problem in graph theory. *Math. Gaz.*, 47:220–223, 1963.

6 R L Graham and J H Spencer. A constructive solution to a tournament problem. *Canad. Math. Bull.*, 14(1):45–48, 1971.

7 R Mehta, I Panageas, and G Piliouras. Natural selection as an inhibitor of genetic diversity: Multiplicative weights updates algorithm and a conjecture of haploid genetics. In *Proc. ITCS*, page 73. ACM, 2015.

8 T Nagylaki. The evolution of multilocus systems under weak selection. *Genetics*, 134(2):627–647, 1993.

9 MA Nowak, A Sasaki, C Taylor, and D Fudenberg. Emergence of cooperation and evolutionary stability in finite populations. *Nature*, 428:646–650, 2004. doi:10.1038/nature02414.

10 T Ohta. Near-neutrality in evolution of genes and gene regulation. *Proc. National Academy of Sciences*, 99(25):16134–16137, 2002. doi:10.1073/pnas.252626899.

11 T Roughgarden and E Tardos. Ch. 17: Introduction to the inefficiency of equilibria. In N Nisan, T Roughgarden, E Tardos, and V V Vazirani, editors, *Algorithmic Game Theory*. Cambridge U Press, 2007.

12 J Von Neumann. Zur theorie der gesellschaftsspiele. *Math. Ann.*, 100:295–320, 1928.

13 B Wu, PM Altrock, L Wang, and A Traulsen. Universality of weak selection. *Physical Review E*, 82:046106, 2010.

14 B Wu, J García, C Hauert, and A Traulsen. Extrapolating weak selection in evolutionary games. *PLoS Comput Biol*, 9(12), 2013. doi:10.1371/journal.pcbi.1003381.

Well-Supported vs. Approximate Nash Equilibria: Query Complexity of Large Games*

Xi Chen[†1], **Yu Cheng**[‡2], **and Bo Tang**[§3]

1 **Columbia University, New York, USA**
xichen@cs.columbia.edu
2 **University of Southern California, Los Angeles, USA**
yu.cheng.1@usc.edu
3 **Oxford University, Oxford, United Kingdom**
tangbonk1@gmail.com

Abstract

In this paper we present a generic reduction from the problem of finding an ϵ-well-supported Nash equilibrium (WSNE) to that of finding an $\Theta(\epsilon)$-approximate Nash equilibrium (ANE), in large games with n players and a bounded number of strategies for each player. Our reduction complements the existing literature on relations between WSNE and ANE, and can be applied to extend hardness results on WSNE to similar results on ANE. This allows one to focus on WSNE first, which is in general easier to analyze and control in hardness constructions.

As an application we prove a $2^{\Omega(n/\log n)}$ lower bound on the randomized query complexity of finding an ϵ-ANE in binary-action n-player games, for some constant $\epsilon > 0$. This answers an open problem posed by Hart and Nisan [23] and Babichenko [2], and is very close to the trivial upper bound of 2^n. Previously for WSNE, Babichenko [2] showed a $2^{\Omega(n)}$ lower bound on the randomized query complexity of finding an ϵ-WSNE for some constant $\epsilon > 0$. Our result follows directly from combining [2] and our new reduction from WSNE to ANE.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases Equilibrium Computation, Query Complexity, Large Games

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.57

1 Introduction

The celebrated theorem of Nash [29] states that every finite game has an equilibrium point. The solution concept of Nash equilibrium (NE) has been tremendously influential in economics and social sciences ever since (e.g. see [24]). The complexity and efficient approximation of NE have been studied intensively during the past decade, and much progress has been made (e.g., see [27, 1, 6, 26, 34, 13, 9, 16, 28, 14, 4, 10, 32, 7, 12, 15, 33, 11, 5]).

In this paper, we study the randomized query complexity of finding an *ϵ-approximate Nash equilibrium* (ANE) in large games, for some *constant* $\epsilon > 0$. Given a game $\mathcal{G}$ with n players and α actions for each player, we index the players by the set $[n] = \{1, \ldots, n\}$ and index the actions by the set $[\alpha] = \{1, \ldots, \alpha\}$. Recall that an ϵ-ANE of $\mathcal{G}$ is a mixed strategy profile $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$, where each $\boldsymbol{x}_i \in [0,1]^\alpha$ sums to 1 and is an ϵ-best response

* This work was done in part while the authors were visiting the Simons Institute.
† Xi Chen is supported in part by NSF grants CCF-1149257 and CCF-1423100.
‡ Yu Cheng is supported in part by Shang-Hua Teng's Simons Investigator Award.
§ Bo Tang is supported in part by ERC grant 321171.


© Xi Chen, Yu Cheng, and Bo Tang;
licensed under Creative Commons License CC-BY

of player i to other players' strategies $\boldsymbol{x}_{-i}$[1] (see Section 2 for the formal definitions). Since the notion of ANE is additive, we always assume that payoff functions of games considered in this paper take values between 0 and 1.

We consider the payoff query model, where an oracle algorithm with unlimited computational power is given an approximation parameter ϵ, the number of players n and the number of actions α in an unknown game $\mathcal{G}$, and needs to find an ϵ-ANE of $\mathcal{G}$. The algorithm has oracle access to the payoff functions of players in $\mathcal{G}$: For each round, the algorithm can adaptively query a pure strategy profile $\boldsymbol{a} \in [\alpha]^n$, and receives the payoff of each player with respect to $\boldsymbol{a}$. We are interested in the number of queries needed by any randomized oracle algorithm for this task. Note that a trivial upper bound is α^n by simply querying all the pure strategy profiles.

1.1 Prior Results and Related Work

The query complexity of (approximate) Nash equilibria and related solution concepts has received considerable attention recently, e.g., see [17, 23, 18, 19, 2, 3, 20, 33]. Below we review results that are most relevant to our work.

The query complexity of (approximate) correlated equilibria (CE) [2] is well understood. For the payoff query model considered here, randomized algorithms exist (e.g., regret-minimizing algorithms [22, 21, 8]) for finding an ϵ-CE using $\text{poly}(1/\epsilon, \alpha, n)$ many queries. It turns out that both randomization and approximation are necessary. [3] showed that every deterministic algorithm that finds an exact CE requires exponentially many queries in n. [23] then showed that the same exponential lower bound holds for any deterministic algorithm for (1/2)-CE and any randomized algorithm for exact CE. For the stronger (expected payoff) query model, where the oracle returns the expected payoffs of any mixed strategy profile [3], [30] and [25] obtained a deterministic algorithm that computes an exact CE in polynomial time using polynomially many queries (both in α and n).

Now turning to the harder, but perhaps more interesting, problem of approximating Nash equilibria under the payoff query model, the deterministic lower bound of [23] for (1/2)-CE directly implies the same bound for (1/2)-ANE, because any ϵ-ANE by definition is an ϵ-CE as well. For the randomized query complexity, Babichenko [2] showed that any randomized algorithm requires $2^{\Omega(n)}$ queries to find an ϵ-well-supported Nash equilibrium (WSNE), in a binary-action, n-player game. Recall that an ϵ-WSNE of a game is a mixed strategy profile $\boldsymbol{x}$ in which the probability of player i playing action j is positive only when action j is an ϵ-best response with respect to $\boldsymbol{x}_{-i}$. By definition, an ϵ-WSNE is also an ϵ-ANE but the inverse is not true. Following a well-known connection between WSNE and ANE [13] (and using random samples to approximate expected payoffs), [2] showed that the same $2^{\Omega(n)}$ bound holds for the randomized query complexity of ϵ-ANE, but only when $\epsilon = O(1/n)$. Before our work, the randomized query complexity of ϵ-ANE in large games remains an open problem when $\epsilon > 0$ is a constant.

In this paper we prove a $2^{\Omega(n/\log n)}$ lower bound on the randomized query complexity of finding an ϵ-ANE, for some constant $\epsilon > 0$. Subsequently, Rubinstein [33] showed a tight $2^{\Omega(n)}$ lower bound on the randomized query complexity of ϵ-ANE, using more sophisticated machinery in coding theory to remove the $\log n$ factor in the exponent. However, our

[1] We use $\boldsymbol{x}_{-i} := (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_n)$ to denote the strategies of players other than i in $\boldsymbol{x}$.

[2] An ϵ-correlated equilibrium is a probability distribution over pure strategy profiles, such that any player unilaterally deviating from strategies drawn from it can increase her expected payoff by no more than ϵ.

[3] Such an oracle can be implemented in polynomial time for many classes of succinct games; see [30].

reduction from ANE to WSNE is much simpler and could be used to obtain hardness results on ANE in other applications (e.g. [31]). It remains an interesting open question if there is a more efficient reduction from WSNE to ANE, without paying the extra factor of $\log n$.

1.2 Our Results

For binary-action, n-player games, we show that $2^{\Omega(n/\log n)}$ queries are required for any randomized algorithm to find an ϵ-ANE, for some constant $\epsilon > 0$. To state the result we use $\mathrm{QC}_p(\mathbf{ANE}(n,\epsilon))$, for some $p > 0$, to denote the smallest T such that there exists a randomized oracle algorithm that uses no more than T queries and outputs an ϵ-ANE with probability at least p, given any unknown binary-action, n-player game. Our main result is the following lower bound on $\mathrm{QC}_p(\mathbf{ANE}(n,\epsilon))$:

▶ **Theorem 1** (Main). *There exist two constants $\epsilon > 0$ and $c > 0$ such that*

$$\mathrm{QC}_p(\mathbf{ANE}(n,\epsilon)) = 2^{\Omega(n/\log n)}, \quad \text{where } p = 2^{-cn/\log n}.$$

Our lower bound answers an open problem posed in [23] and in [2]. Our result shows that, in terms of their query complexities, finding an ϵ-ANE is almost as hard as finding an ϵ-WSNE in a large game, even for constant $\epsilon > 0$. It also directly implies the following corollary regarding the rate of convergence of k-queries dynamics (see [2] for the definition).

▶ **Corollary 2.** *There exist constants $\epsilon, c > 0$ such that no k-queries dynamic can converge to an ϵ-ANE in $2^{\Omega(n/\log n)}/k$ steps with probability at least $2^{-cn/\log n}$ for every binary-action and n-player game.*

Our proof of Theorem 1 relies on a polynomial-time reduction [4] from the problem of finding an ϵ-WSNE to that of finding an $(\epsilon' = \Omega(\epsilon))$-ANE in a *succinct game* with a fixed number of actions. As defined in [30], an α-action succinct game is a pair (n, U), where n is the number of players and U is a (multi-output) Boolean circuit that, given a pure strategy profile $\boldsymbol{a} \in [\alpha]^n$ (encoded in binary), outputs the payoffs of all n players with respect to $\boldsymbol{a}$.

▶ **Theorem 3.** *Let $\epsilon \geq 0$ and $\alpha \in \mathbb{N}$ be two constants. The problem of finding an ϵ-WSNE is polynomial-time reducible to that of finding an $\epsilon/(4\alpha)$-ANE, both in α-action succinct games.*

1.3 Approximate vs. Well-Supported Nash Equilibria

Let $\mathrm{QC}_p(\mathbf{WSNE}(n,\epsilon))$ denote the smallest T such that there exists a randomized oracle algorithm that uses no more than T queries and outputs an ϵ-WSNE with probability at least p, given any unknown binary-action, n-player game. Babichenko [2] showed that

▶ **Theorem 4** ([2]). *There exist constants $\epsilon, c > 0$ such that*

$$\mathrm{QC}_p(\mathbf{WSNE}(n,\epsilon)) = 2^{\Omega(n)}, \quad \text{where } p = 2^{-cn}.$$

Given Theorem 4, the same exponential lower bound follows directly for the randomized query complexity of ϵ-ANE, for certain small enough constant $\epsilon > 0$, if

[4] Recall that a polynomial-time reduction from a total search problem A to a total search problem B is a pair (f, g) of polynomial-time computable functions such that: 1) for every input instance x of A, $f(x)$ is an input instance of B; and 2) for every solution y to $f(x)$ in B, $g(y)$ is a solution to x in A.

Given oracle access to $\mathcal{G}$ and any ϵ'-ANE of $\mathcal{G}$, where $\epsilon' = c(\alpha) \cdot \epsilon$ for some constant $c > 0$ that only depends on α, there is a query-efficient procedure that outputs an ϵ-WSNE of $\mathcal{G}$.

However, the best such procedure known is the following result from [13]. The parameters are subsequently improved in [2], where the number of queries needed is also analyzed:

Given oracle access to $\mathcal{G}$ and any $\epsilon^2/(16n)$-ANE of $\mathcal{G}$, there is a procedure that outputs an ϵ-WSNE of $\mathcal{G}$ using poly$(\alpha, n, 1/\epsilon)$ payoff queries, where n denotes the number of players.

The procedure is very natural: For each player, reallocate probabilities on actions with a relatively low expected payoff to a best-response action. Using Theorem 4, such a procedure implies the same exponential lower bound for ϵ-ANE [2] but only when ϵ is $O(1/n)$.

Before our work, no better procedure is known. By definition, an ANE poses a slightly weaker condition on each player compared to that of a WSNE. More specifically, given the mixed strategies of other players $\boldsymbol{x}_{-i}$, for an ϵ-WSNE, $\boldsymbol{x}_i$ must be supported on actions that are ϵ-best responses to $\boldsymbol{x}_{-i}$, while in an ϵ-ANE, $\boldsymbol{x}_i$ can be any mixed strategy that yields an overall ϵ-best response to $\boldsymbol{x}_{-i}$. For example, $\boldsymbol{x}_i$ may allocate $1 - \epsilon$ probability on best-response actions while putting ϵ probability on any other actions. This makes WSNE much easier to analyze and control in hardness reductions, which is why it played a critical role in characterizing the complexity of Nash equilibria, starting with the work of [13], later in [9] and subsequent works. The reason that Babichenko's lower bound (Theorem 4) does not hold for ϵ-ANE is that, if every player places a tiny probability on a suboptimal action, in aggregate there are always some players who play suboptimally, which makes the outcome quite unpredictable.

1.4 Our Approach

We prove Theorem 1 via a *query-efficient* reduction from the problem of finding an ϵ-WSNE to that of finding an $\Theta(\epsilon)$-ANE:

Given any α-action, n-player game $\mathcal{G}$ and any parameter $\epsilon > 0$, one can define a new α-action game $\mathcal{G}'$ with a slightly larger set of $O(\alpha^2 \log(n/\epsilon) \cdot n)$ players such that

1. *To answer each payoff query on $\mathcal{G}'$, it suffices to make αn payoff queries on $\mathcal{G}$;*
2. *There is a procedure that, given any ϵ-ANE $\boldsymbol{x}$ of $\mathcal{G}'$, outputs a $(4\alpha\epsilon)$-WSNE $\boldsymbol{y}$ of $\mathcal{G}$, with no payoff oracle access to $\mathcal{G}$ or $\mathcal{G}'$.*

Our reduction is presented in Section 3. Theorem 1 then follows immediately from the lower bound of [2] on the randomized query complexity of WSNE (in Theorem 4). Theorem 3 follows from the fact that: 1) the payoff entries of $\mathcal{G}'$ are easy to compute; and 2) the procedure to obtain $\boldsymbol{y}$ from $\boldsymbol{x}$ runs in time polynomial in the length of the binary representation of $\boldsymbol{x}$, when the number of actions α is bounded. We first give the intuition behind our reduction.

Recall that in the procedure of [13] and [2], an ϵ-WSNE is obtained from an ϵ'-ANE with $\epsilon' = \epsilon^2/(16n)$ by reallocating probabilities on actions with relatively low expected payoff (formally, actions with payoff $\Omega(\epsilon)$ lower than the best response) to best-response actions. From the definition of ANE, no player can have probability more than $O(\epsilon'/\epsilon) = O(\epsilon/n)$ on actions with low payoff in any ϵ'-ANE. Thus, the procedure changes the expected payoff of each player on each action by at most $n \cdot O(\epsilon/n) = O(\epsilon)$ since it changes the mixed strategy of each player by $O(\epsilon/n)$. It follows that the new mixed strategy profile is an ϵ-WSNE. The blow up of a factor of n from ϵ' to ϵ is precisely due to the cumulative impact on a player's expected payoff imposed by small changes to all other players' mixed strategies.

Our reduction from WSNE to ANE overcomes this obstacle by constructing from $\mathcal{G}$ a new and slightly larger game $\mathcal{G}'$ with $O(n \log n)$ players, where each player i in the original n-player game $\mathcal{G}$ is simulated by a group of $O(\log n)$ players indexed by (i, j) in the new game $\mathcal{G}'$, and we use the *majority* strategy in group i to decide the strategy of player i in the original game. The payoff function of the player (i, j) in $\mathcal{G}'$ is exactly the same as that of player i in $\mathcal{G}$, but is now defined with respect to the *aggregate* action (by plurality voting) of each group of players in $\mathcal{G}'$.

We show that an ϵ-WSNE of $\mathcal{G}$ can be recovered from any ϵ'-ANE of $\mathcal{G}'$, where $\epsilon' = \Omega(\epsilon)$, by (1) computing the distribution of the majority action of each group and (2) truncating the small entries in each distribution. Intuitively, by focusing on the aggregate behavior of each group of $O(\log n)$ *independent* players in $\mathcal{G}'$, we make sure that the mixed strategies obtained from Step (1) are highly concentrated on actions with close-to-best expected payoffs, and actions with low payoffs can only appear as the majority action of a group with probability $O(\epsilon/n)$. Therefore, in Step (2) we only need to truncate entries with probability $O(\epsilon/n)$, and the remaining positive entries would correspond to close-to-best actions. We can also control the effect of this truncation at the same time, because when the number of actions is bounded, the aggregate behavior of each group changes by at most $O(\epsilon/n)$, which allows us to show that the result is an ϵ-WSNE of the original game $\mathcal{G}$.

1.5 Organization

The rest of the paper is organized as follows. We first give formal definitions of ANE and WSNE in Section 2. In Section 3 we present the reduction from WSNE to ANE for large games, and then use it to prove Theorem 1 and Theorem 3 in Section 4.

2 Preliminaries

A game $\mathcal{G}$ is a triple $(n, \alpha, \boldsymbol{u})$, where n is the number of players, α is the number of actions for each player, and $\boldsymbol{u} = (u_1, \ldots, u_n)$ are the payoff functions, one for each player. We always use $[n] = \{1, \ldots, n\}$ to denote the set of players and $[\alpha] = \{1, \ldots, \alpha\}$ to denote the set of actions for each player. Since we are interested in additive approximations, each u_i maps $[\alpha]^n$ to $[0, 1]$.

Let Δ_α denote the set of probability distributions over $[\alpha]$. A mixed strategy profile of $\mathcal{G}$ is then a tuple $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ of mixed strategies, where $\boldsymbol{x}_i \in \Delta_\alpha$ denotes the mixed strategy of player i. Given $\boldsymbol{x}$, we use $\boldsymbol{x}_{-i}$ to denote the tuple of mixed strategies of all players other than i. As a shorthand, we write $u_i(\boldsymbol{x})$ to denote the expected payoff of player i with respect to $\boldsymbol{x}$, and write $u_i(a, \boldsymbol{x}_{-i})$ to denote the expected payoff of player i playing action $a \in [\alpha]$ with respect to $\boldsymbol{x}_{-i}$:

$$u_i(\boldsymbol{x}) = \mathop{\mathbb{E}}_{\boldsymbol{a} \sim \boldsymbol{x}} [u_i(\boldsymbol{a})] \quad \text{and} \quad u_i(a, \boldsymbol{x}) = \mathop{\mathbb{E}}_{\boldsymbol{b} \sim \boldsymbol{x}_{-i}} \big[u_i(a, \boldsymbol{b})\big].$$

Next we define approximate and well-supported Nash equilibria.

▶ **Definition 5.** Given $\epsilon > 0$, an ϵ-approximate Nash equilibrium of an α-action and n-player game $\mathcal{G}(n, \alpha, \boldsymbol{u})$ is a mixed strategy profile $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ such that for every player $i \in [n]$:

$$u_i(\boldsymbol{x}) \geq u_i(a', \boldsymbol{x}_{-i}) - \epsilon, \quad \text{for all } a' \in [\alpha].$$

▶ **Definition 6.** Given $\epsilon > 0$, an ϵ-well-supported Nash equilibrium of $\mathcal{G}(n, \alpha, \boldsymbol{u})$ is a mixed strategy profile $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ such that for all $i \in [n]$ and action a in the support of $\boldsymbol{x}_i$:

$$u_i(a, \boldsymbol{x}_{-i}) \geq u_i(a', \boldsymbol{x}_{-i}) - \epsilon, \quad \text{for all } a' \in [\alpha].$$

Finally, we give a formal definition of succinct games [30].

▶ **Definition 7.** An α-action succinct game is a pair (n, U), where n is the number of players and U is a (multi-output) Boolean circuit that, given any pure strategy profile $\boldsymbol{a} \in [\alpha]^n$ (encoded in binary), outputs the payoffs of all n players with respect to $\boldsymbol{a}$ in the game. Note that the input size of (n, U) is the size of the circuit U.

3 A Reduction from WSNE to ANE

Given an α-action, n-player game $\mathcal{G}(n, \alpha, \boldsymbol{u})$ and a parameter $\epsilon \in (0, 1)$, we now define a new game $\mathcal{G}'(sn, \alpha, \boldsymbol{u}')$ with sn players, where $s = 2\alpha^2 \cdot \lceil \ln(n/\epsilon) \rceil$. We prove that given an ϵ-ANE $\boldsymbol{x}$ of the new game $\mathcal{G}'$, one can compute a $(4\alpha\epsilon)$-WSNE $\boldsymbol{y}$ of $\mathcal{G}$ without making any payoff queries to $\mathcal{G}$ or $\mathcal{G}'$.

For each player $i \in [n]$ in $\mathcal{G}$, we introduce a group of s players in $\mathcal{G}'$, indexed by (i, j) with $j \in [s]$, and use $u'_{i,j}$ to denote the payoff function of player (i, j). Given any pure strategy profile $\boldsymbol{a} = (a_{i,j} : i \in [n], j \in [s])$, we define the payoff $u'_{i,j}(\boldsymbol{a})$ of player (i, j) as follows. First, for each $i \in [n]$, let $\bar{a}_i \in [\alpha]$ denote the *majority* action played by the i-th group (players (i, j), $j \in [s]$) in the pure strategy profile $\boldsymbol{a}$ (break ties by choosing the action with the smallest index). Write $\bar{\boldsymbol{a}} = (\bar{a}_1, \ldots, \bar{a}_n)$. Next, the payoff of player (i, j) under $\boldsymbol{a}$ is defined as

$$u'_{i,j}(\boldsymbol{a}) = u_i(a_{i,j}, \bar{\boldsymbol{a}}_{-i}). \tag{1}$$

This completes the definition of $\mathcal{G}'$. The next lemma follows from the definition.

▶ **Lemma 8.** *To answer a payoff query on $\mathcal{G}'$, it suffices to make αn queries on $\mathcal{G}$.*

Proof. By the definition of $\mathcal{G}'$, $u'_{i,j}(\boldsymbol{a})$'s for all (i, j), are determined by

$$\big(u_i(a', \bar{\boldsymbol{a}}_{-i}) : i \in [n], a' \in [\alpha]\big),$$

for which αn payoff queries on $\mathcal{G}$ suffice. ◀

We conclude our reduction by proving the following lemma:

▶ **Lemma 9.** *Given any ϵ-ANE $\boldsymbol{x}$ of $\mathcal{G}'$, one can compute a $(4\alpha\epsilon)$-WSNE $\boldsymbol{y}$ of G without making any payoff queries on $\mathcal{G}$ or $\mathcal{G}'$. Moreover, when α is a constant, the computation of $\boldsymbol{y}$ from $\boldsymbol{x}$ can be done in time polynomial in the number of bits needed in the binary representation of $\boldsymbol{x}$ and $1/\epsilon$.*

Proof. Let $\boldsymbol{x} = (\boldsymbol{x}_{i,j})$ be an ϵ-ANE of $\mathcal{G}'$. For each group i and action $k \in [\alpha]$, let

$$\bar{x}_{i,k} = \Pr_{\boldsymbol{a} \sim \boldsymbol{x}} [\bar{a}_i = k]. \tag{2}$$

Recall that $\bar{a}_i$ is the majority action played by players (i, j), $j \in [s]$, in the pure strategy profile $\boldsymbol{a}$. By definition, each $\bar{\boldsymbol{x}}_i = (\bar{x}_{i,1}, \ldots, \bar{x}_{i,\alpha})$ is a probability distribution over $[\alpha]$.

Next, we define a mixed strategy $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)$ of $\mathcal{G}$, and show that $\boldsymbol{y}$ is a $(4\alpha\epsilon)$-WSNE. We zero out entries smaller than ϵ/n in $\bar{\boldsymbol{x}}_i$ and rescale it, formally

$$c_{i,k} = \begin{cases} \bar{x}_{i,k} & \text{if } \bar{x}_{i,k} \le \epsilon/n \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad y_{i,k} = \frac{\bar{x}_{i,k} - c_{i,k}}{1 - \sum_{j \in [\alpha]} c_{i,k}}. \tag{3}$$

It is easy to verify that $\boldsymbol{y}_i = (y_{i,1}, \ldots, y_{i,\alpha})$ is indeed a probability distribution over $[\alpha]$.

Now assume for contradiction that $\boldsymbol{y}$ is not a $(4\alpha\epsilon)$-WSNE, i.e. for some player $i \in [n]$ there exists an action $\ell \in [\alpha]$ such that $y_{i,\ell} > 0$ but

$$\max_{k\in[\alpha]} u_i(k, \boldsymbol{y}_{-i}) > u_i(\ell, \boldsymbol{y}_{-i}) + 4\alpha\epsilon. \tag{4}$$

But note that, the total variation distance between $\bar{\boldsymbol{x}}_j$ and $\boldsymbol{y}_j$ for each $j \in [n]$ is at most $\alpha\epsilon/n$. So by coupling and applying union bound, we have that

$$\left|u_i(k, \bar{\boldsymbol{x}}_{-i}) - u_i(k, \boldsymbol{y}_{-i})\right| \le (n-1)\cdot(\alpha\epsilon/n) < \alpha\epsilon, \quad \text{for all } k \in [\alpha]. \tag{5}$$

It then follows from (4) and (5) that

$$\max_{k\in[\alpha]} u_i(k, \bar{\boldsymbol{x}}_{-i}) > u_i(\ell, \bar{\boldsymbol{x}}_{-i}) + 2\alpha\epsilon. \tag{6}$$

By the definition (1) of the payoff function $u'_{i,j}$, we have

$$u'_{i,j}(k, \boldsymbol{x}_{-i}) = u_i(k, \bar{\boldsymbol{x}}_{-i}), \quad \text{for all } j \in [s] \text{ and } k \in [\alpha]. \tag{7}$$

Combining (6) and (7), we have that for every player (i,j), $j \in [s]$:

$$\max_{k\in[\alpha]} u'_{i,j}(k, \boldsymbol{x}_{-i}) - u'_{i,j}(\ell, \boldsymbol{x}_{-i}) \ge 2\alpha\epsilon.$$

Since $\boldsymbol{x}$ is an ϵ-ANE of $\mathcal{G}'$, $x_{i,j,\ell} \le 1/(2\alpha)$. By Hoeffding bound and our choice of s,

$$\begin{aligned}\bar{x}_{i,\ell} &= \Pr\left[\ell \text{ is the majority action among players } (i,j),\ j \in [s]\right] \\ &\le \Pr\left[\text{the number of players } (i,j) \text{ playing } \ell \text{ is at least } s/\alpha\right] \le e^{-s/(2\alpha^2)} \le \epsilon/n.\end{aligned}$$

By (3), this implies that $y_{i,\ell} = 0$, which contradicts our assumption and proves that $\boldsymbol{y}$ is indeed a $(4\alpha\epsilon)$-WSNE of $\mathcal{G}$.

It is clear that from the definition of $\boldsymbol{y}$, the computation of $\boldsymbol{y}$ from $\boldsymbol{x}$ does not require any payoff queries. For the running time, when α is a constant, to compute $\bar{x}_{i,k}$ in (2) one needs to go through

$$\alpha^s = \alpha^{2\alpha^2\cdot\lceil\ln(n/\epsilon)\rceil} = (n/\epsilon)^{O(1)}$$

many pure strategy profiles of players (i,j), $j \in [s]$. Therefore $\boldsymbol{y}$ can be computed in time polynomial in the number of bits needed in the binary representation of $\boldsymbol{x}$ and $1/\epsilon$. ◀

4 Proofs of Theorems 1 and 3

We use our query-efficient reduction to prove Theorem 1 and Theorem 3.

Proof of Theorem 1. By Theorem 4 there exist constants $\epsilon', c' > 0$ such that

$$\mathrm{QC}_{p'}(\mathbf{WSNE}(n', \epsilon)) = 2^{\Omega(n')}, \quad \text{where } p' = 2^{-c'n'}.$$

Let $n = 8n' \cdot \lceil\ln(n'/\epsilon')\rceil$ and $\epsilon = 8\epsilon'$. It follows from Lemma 8 and Lemma 9 that

$$\mathrm{QC}_{p'}(\mathbf{ANE}(n, \epsilon)) \ge \mathrm{QC}_{p'}(\mathbf{WSNE}(n', \epsilon)) = 2^{\Omega(n')}.$$

The theorem then follows from $n' = \Omega(n/\log n)$. ◀

Proof of Theorem 3. Using Lemma 9, it suffices to prove that, given any α-action succinct game $\mathcal{G} = (n, U)$, we can construct in polynomial time a Boolean circuit U' that implements the payoff functions of players in $\mathcal{G}'$. This can be done by following the definition of $\mathcal{G}'$ in the previous section since the payoffs of a pure strategy profile $\boldsymbol{a}$ in $\mathcal{G}'$ only depends (in a straight-forward fashion) on the payoffs of $\alpha n = O(n)$ easy-to-compute profiles of $\mathcal{G}$. ◀

References

1. T. Abbott, D. Kane, and P. Valiant. On the complexity of two-player win-lose games. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 113–122, 2005.
2. Y. Babichenko. Query complexity of approximate Nash equilibria. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 535–544, 2014.
3. Y. Babichenko and S. Barman. Query complexity of correlated equilibrium. *ACM Transactions on Economics and Computation*, 3(4):22:1–22:9, 2015.
4. Y. Babichenko, C.H. Papadimitriou, and A. Rubinstein. Can almost everybody be almost happy? PCP for PPAD and the inapproximability of Nash. In *Proceedings of the 7th Annual Innovations in Theoretical Computer Science*, 2015.
5. Y. Babichenko and A. Rubinstein. Communication complexity of approximate Nash equilibria. *ArXiv e-prints*, 2016.
6. I. Bárány, S. Vempala, and A. Vetta. Nash equilibria in random games. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 123–131, 2005.
7. S. Barman. Approximating Nash equilibria and dense bipartite subgraphs via an approximate version of Caratheodory's theorem. In *Proceedings of the 47th Annual ACM on Symposium on Theory of Computing*, pages 361–369, 2015.
8. A. Blum and Y. Monsour. *From External to Internal Regret*. University of Chicago Press, 2007.
9. X. Chen, X. Deng, and S.-H. Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM*, 56(3):1–57, 2009.
10. X. Chen, D. Durfee, and A. Orfanou. On the complexity of Nash equilibria in anonymous games. In *Proceedings of the 46th ACM Symposium on Theory of Computing*, pages 381–390, 2015.
11. Y. Cheng, I. Diakonikolas, and A. Stewart. Playing anonymous games using simple strategies. In *Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2017. To appear.
12. C. Daskalakis, A. De, G. Kamath, and C. Tzamos. A size-free CLT for Poisson multinomials and its applications. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing*, pages 1074–1086, 2016.
13. C. Daskalakis, P.W. Goldberg, and C.H. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39(1), 2009.
14. C. Daskalakis and C.H. Papadimitriou. Approximate Nash equilibria in anonymous games. *Journal of Economic Theory*, 156:207–245, 2015.
15. I. Diakonikolas, D.M. Kane, and A. Stewart. The Fourier transform of Poisson multinomial distributions and its algorithmic applications. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing*, pages 1060–1073, 2016.
16. K. Etessami and M. Yannakakis. On the complexity of Nash equilibria and other fixed points. *SIAM Journal on Computing*, 39(6):2531–2597, 2010.
17. J. Fearnley, M. Gairing, P. Goldberg, and R. Savani. Learning equilibria of games via payoff queries. In *Proceedings of the 14th ACM Conference on Electronic Commerce*, pages 397–414, 2013.
18. J. Fearnley and R. Savani. Finding approximate Nash equilibria of bimatrix games via payoff queries. In *Proceedings of the 15th ACM Conference on Economics and Computation*, pages 657–674, 2014.
19. P.W. Goldberg and A. Roth. Bounds for the query complexity of approximate equilibria. In *Proceedings of the 15th ACM Conference on Economics and Computation*, pages 639–656, 2014.

20 P.W. Goldberg and S. Turchetta. Query complexity of approximate equilibria in anonymous games. In *Proceedings of the 11th Conference on Web and Internet Economics*, 2015.

21 S. Hart. Adaptive heuristics. *Econometrica*, 73(5):1401–1430, 2005.

22 S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

23 S. Hart and N. Nisan. The query complexity of correlated equilibria. In *Proceedings of the 6th International Symposium on Algorithmic Game Theory*, 2013.

24 C.A. Holt and A.E. Roth. The Nash equilibrium: A perspective. *Proceedings of the National Academy of Sciences*, 101(12):3999–4002, 2004.

25 A.X. Jiang and K. Leyton-Brown. Polynomial-time computation of exact correlated equilibrium in compact games. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, 2011.

26 R. Kannan and T. Theobald. Games of fixed rank: A hierarchy of bimatrix games. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1124–1132, 2007.

27 R.J. Lipton, E. Markakis, and A. Mehta. Playing large games using simple strategies. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, pages 36–41, 2003.

28 R. Mehta. Constant rank bimatrix games are PPAD-hard. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 545–554, 2014.

29 J.F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.

30 C.H. Papadimitriou and T. Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM*, 55(3), 2008.

31 T. Roughgarden and O. Weinstein. On the communication complexity of approximate fixed points. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, pages 229–238, 2016.

32 A. Rubinstein. Inapproximability of Nash equilibrium. In *Proceedings of the 46th ACM Symposium on Theory of Computing*, pages 409–418, 2015.

33 A. Rubinstein. Settling the complexity of computing approximate two-player Nash equilibria. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, pages 258–265, 2016.

34 H. Tsaknakis and P.G. Spirakis. An optimization approach for approximate Nash equilibria. In *Proceedings of the 3rd International Workshop on Internet and Network Economics*, pages 42–56, 2007.

Metatheorems for Dynamic Weighted Matching

Daniel Stubbs[*1] **and Virginia Vassilevska Williams**[†2]

1 **Computer Science Department, Stanford University, USA**
dstubbs@stanford.edu
2 **Computer Science Department, Stanford University, USA**
virgi@cs.stanford.edu

Abstract

We consider the maximum weight matching (MWM) problem in dynamic graphs. We provide two reductions. The first reduces the dynamic MWM problem on m-edge, n-node graphs with weights bounded by N to the problem with weights bounded by $(n/\varepsilon)^2$, so that if the MWM problem can be α-approximated with update time $t(m, n, N)$, then it can also be $(1+\varepsilon)\alpha$-approximated with update time $O(t(m, n, (n/\varepsilon)^2)\log^2 n + \log n \log\log N))$. The second reduction reduces the dynamic MWM problem to the dynamic maximum cardinality matching (MCM) problem in which the graph is unweighted. This reduction shows that if there is an α-approximation algorithm for MCM with update time $t(m, n)$ in m-edge n-node graphs, then there is also a $(2+\varepsilon)\alpha$-approximation algorithm for MWM with update time $O(t(m, n)\varepsilon^{-2}\log^2 N)$. We also obtain better bounds in our reductions if the ratio between the largest and the smallest edge weight is small. Combined with recent work on MCM, these two reductions substantially improve upon the state-of-the-art of dynamic MWM algorithms.

1998 ACM Subject Classification F2.2 Nonnumerical Algorithms and Problems

Keywords and phrases dynamic algorithms, maximum matching, maximum weight matching

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.58

1 Introduction

The maximum matching problem is one of the most basic problems in algorithms. Starting from the 1950s, a series of influential papers (among others, [15, 13, 20, 23, 9, 14, 17, 19]) culminated in very efficient algorithms for the problem in n node m edge graphs. There are two versions of the problem: the unweighted version in which one needs to return a maximum cardinality matching (MCM), and the weighted version, in which the input graph has weights on its edges and one seeks a matching of maximum weight (MWM).

For MCM in general graphs, the best known algorithms are the Micali-Vazirani [18] $O(m\sqrt{n})$ algorithm and the $O(n^\omega)$ time algorithm of Mucha and Sankowski [19] (see also [12, 23, 20]), where $\omega < 2.373$ is the matrix multiplication exponent [26, 16]. For the special case of bipartite graphs, a recent breakthrough result of Madry [17] achieved a runtime of $\tilde{O}(m^{10/7})$. For the maximum weight matching problem, the best known running times are $O(m\sqrt{n}\log(nN))$ [9, 10] and $\tilde{O}(Nn^\omega)$ [25] where N is the largest edge weight.

Graphs in applications, however, are dynamic by nature. Edges and vertices fail and new ones are introduced. Because of this, more resilient, *dynamic* algorithms are desired. Such algorithms can update the solution efficiently when an edge insertion or deletion occurs.

* Supported by an NSF Graduate Fellowship.
† Supported by NSF Grants CCF-1417238, CCF-1528078 and CCF-1514339, and BSF Grant BSF:2012338.


© Daniel Stubbs and Virginia Vassilevska Williams;
licensed under Creative Commons License CC-BY

Maintaining an exact solution to the maximum matching problem, however, turns out to be a difficult problem. Utilizing fast matrix multiplication, Sankowski [24] gave an algorithm with update time $O(N^{2.495}n^{1.495})$ that can maintain the weight of a maximum weight matching; his algorithm is also the best known in the unweighted case when $N = 1$. Abboud and Vassilevska Williams [1] showed in a formal sense that the use of fast matrix multiplication is inherent even for dynamic bipartite MCM, as fast enough updates would imply a faster Boolean matrix multiplication algorithm.

Because exact dynamic algorithms seem to be doomed to be inefficient, the majority of research on dynamic matching has been on maintaining approximate matchings. Following work by Onak and Rubinfeld [22], Baswana et al. [3] obtained a randomized fully dynamic algorithm that maintains a maximal (and hence 2-approximate) matching with expected amortized update time $O(\log n)$. Obtaining a fast deterministic algorithm has been a much more difficult task. Following Neiman and Solomon [21], Gupta and Peng [11] showed how to obtain a $(1+\varepsilon)$-approximation algorithm to the MCM with update time $O(\sqrt{m}/\varepsilon^2)$. Bernstein and Stein [5, 4] developed dynamic algorithms that achieve a $(3/2 + \epsilon)$-approximation in amortized update time $O(m^{1/4}\epsilon^{-2.5})$. Just recently Bhattacharya et al. [7] gave a very fast deterministic algorithm: for every $\epsilon > 0$, their algorithm achieves a $(2 + \epsilon)$-approximation algorithm with update time $\text{polylog}(n, 1/\epsilon)$.

For maintaining an approximate MWM, only two results are known. Anand et al. [2] obtained a randomized 4.911-approximation to the MWM in $O(\log n \log C + \log N)$ expected amortized update time[1], where N is the maximum edge weight and C is the ratio between N and the lowest edge weight. Gupta and Peng [11] obtained a $(1 + \varepsilon)$ approximation in deterministic worst case $O(\sqrt{m}\varepsilon^{-2-O(1/\varepsilon)} \log N)$ update time, where N is the largest edge weight in the graph.

There are two main questions that emerge:

1. *Can one get improved dynamic algorithms for MWM?* E.g., is there a faster than $O(\sqrt{m})$ deterministic worst case update algorithm for any constant approximation ratio? Is there a polylogarithmic update time dynamic algorithm with a better than 4.911-approximation guarantee? Can one use the recent techniques developed for dynamic MCM algorithms here?
2. All known dynamic MWM algorithms have update times that depend logarithmically on N. If N can be exponential in n, this causes a polynomial overhead in the running time. *Can the dependence on N be decreased?*
   If the algorithm had to be exact, then clearly the algorithm has to read the entire edge weights, so it is natural to have a dependence on $\log N$. However, here the answers can be approximate, so perhaps one can get away with reading only a few of the bits of the weight. If this is possible, then the algorithm would be fast even if the edge weights were exponential in the number of vertices. In particular, a truly polylogarithmic in n update time for approximate MWM would be possible.

1.1 Our results

In this paper we address both questions above.

Our first contribution is an efficient black-box reduction from dynamic approximate MWM to dynamic approximate MCM. Applying this reduction, we are able to improve the

[1] The update time stated in [2] is $O(\log n \log C)$, however, taking into account reading the edge weights on edge insertion actually adds a $\log N$ to the running time.

current best algorithms for dynamic MWM. The second contribution is a black-box reduction from dynamic approximate MWM on an exponential weight range to the same problem on a polynomial weight range, giving rise to the first algorithm for exponential range dynamic approximate MWM with truly polylogarithmic update time.

Let C be the ratio between the largest edge weight N and smallest edge weight L. Let n be the number of nodes and m the number of edges. The two theorems we prove are as follows:

▶ **Theorem 1.** *Let A be a dynamic algorithm that maintains an α-approximate MCM with update time $t(m,n)$. Then, for all $\varepsilon > 0$, there is a dynamic algorithm that maintains a $2\alpha \cdot (1+\varepsilon)$-approximate MWM with update time $O(t(m,n)\varepsilon^{-2}\log^2 C + \log N)$.*

If the original MCM algorithm was fully dynamic, so is the MWM algorithm, and if the original MCM algorithm is partially dynamic (incremental or decremental), so is the MWM one. If the MCM update time was worst case, then so is the MWM one, and if the original one was deterministic, so is the MWM one.

▶ **Theorem 2.** *Suppose that there is an algorithm that maintains an α-approximate MWM with update time $T(m,n,N)$. Then, for every constant $\varepsilon > 0$, one can convert it into an algorithm that maintains an $\alpha \cdot (1+\varepsilon)$-approximate MWM with asymptotic update time*

$$T(m,n,n^2\varepsilon^{-2})\log^2 n + \log n \log\log C + \log\log N.$$

The new algorithm enjoys the same properties as the old one (worst-case, deterministic etc).

Composing Theorems 1 and 2 we get that a $t(m,n)$-update time α-approximate MCM algorithm can be converted into a $(2+\varepsilon)\alpha$-approximate MWM algorithm with update time $O(t(m,n)\varepsilon^{-2}\log^4 n + \log n \log\log C + \log\log N)$.[2]

Applying Theorems 1 and 2 to the algorithms of Baswana et al. [3], Bhattacharya et al. [6, 7] and Bernstein and Stein [5, 4], we obtain the following immediate corollary.

▶ **Corollary 3.** *The following dynamic algorithms exist for MWM for all $\varepsilon > 0$:*

- *a $(4+\varepsilon)$-approximation with deterministic worst case update time $O(\ poly(\log n, 1/\varepsilon) + \log n \log\log C + \log\log N)$*
- *a $(4+\varepsilon)$-approximation with expected amortized update time $O(\varepsilon^{-2}\log^5 n + \log n \log\log C + \log\log N)$, and*
- *a $(3+\varepsilon)$-approximation in deterministic amortized update time $O(m^{1/4}\varepsilon^{-4.5}\log^4 n + \log n \log\log C + \log\log N)$.*

The above Corollary significantly improves upon the state of the art of dynamic MWM algorithms. In particular, it presents the first *truly polylogarithmic* update time for approximate dynamic MWM in the case when the edge weights can be exponential in n. Note that since our algorithm works on subsets of the edges of the original graph, we can derive dynamic weighted matching algorithms from dynamic maximum cardinality matching algorithms for *special classes of graphs* closed under edge deletions, such as the low-arboricity result of Neiman and Solomon [21].

A big advantage of our meta-algorithms is that they are simple, clean and combinatorial. Thus, if the original dynamic MCM algorithm is practical, then the MWM algorithms resulting from our theorems would likely also be practical.

[2] The composition gives this runtime for any $\varepsilon > 1/n^3$. Of course, if $\varepsilon \leq 1/n^3$, the trivial algorithm that recomputes the matching from scratch has update time $O(1/\varepsilon)$.

1.1.1 Overview of the reduction from dynamic MWM to MCM

Here we give an overview of our approach to proving Theorem 1. Our starting point is a result by Crouch and Stubbs [8] that reduced MWM to MCM in the streaming setting. This reduction shows how to reduce approximate MWM to a small number of instances of approximate MCM, but does not show how to maintain the output approximate MWM efficiently under arbitrary edge updates. Here we show how to make the reduction work in the dynamic setting.

Suppose that we are given a graph with integer edge weights between L and N and we want to compute an approximate MWM; let $C = N/L$. The basic idea of the Crouch and Stubbs reduction is to take maximal matchings from weight-threshold-based subgraphs of the underlying graph, and then merge the resulting maximal matchings together greedily. In particular, one computes an approximate MCM on all edges of weight at least $(1+\epsilon)^i$, for $i \in \{\lfloor(\log_{1+\epsilon} L\rfloor, \lfloor\log_{1+\epsilon} L\rfloor + 1, \ldots, \lfloor\log_{1+\epsilon} N\rfloor\}$. One produces the output matching from the approximate MCMs by including all edges from the matching in the weight class with the highest threshold, and then adding edges in descending order of the height of their weight class, as long as they are node-disjoint from the edges added so far.

Since we have $O(\epsilon^{-1} \log C)$ different weight classes, maintaining the approximate MCMs only incurs a small overhead, over any dynamic matching algorithm for approximate MCM. The only remaining concern, then, is how much time it takes to maintain the output matching by merging these together. This is our contribution on top of the result of Crouch and Stubbs: we show that the greedy merge of the MCMs can be *updated* efficiently.

We begin by assuming that we have approximate maximum cardinality matchings for each of the weight classes and an output matching M constructed *statically* by merging these matchings greedily in decreasing order of class height. When an update to the underlying graph occurs, it might cause some number of the edges in each of the matchings to change. This number in a weight class is at most the update time of the corresponding MCM data structure. Assuming that the dynamic algorithm for approximate MCM has update time $T(m, n)$, the number of changed edges in any one of the matchings is at most $T(m, n)$.

When edges change in an approximate MCM, we fold those changes into M one level at a time, in order of decreasing class height. If an edge in the output matching was deleted, we mark its endpoints as "newly free," so that we can look for new edges that cover those endpoints in lower classes – note that we needn't check the higher classes, since any such edge would have precluded the just-deleted edge from being in the output matching. If an edge was newly added, we add it to the output matching iff neither of its endpoints is covered by a higher class edge; if any edges are deleted in the process, we mark their endpoints as "newly free" and check for edges that cover these newly free nodes in lower classes, as before.

The process of checking for edges to cover a newly free node v is simple: in each lower class, as we're rolling in the changes to the current MCM, we also check whether v is matched in the updated MCM. If it is and its match u is not covered by an edge in the current output matching from a higher class, we add (u, v) to the output matching and v is no longer newly free. If u were already matched in the output matching to some node via a less exclusive edge e, we remove e from the output matching and make its other endpoint newly free.

The main part of the efficiency argument is as follows. The changes in the MCMs cause at most $O(T(m, n)\varepsilon^{-1} \log C)$ edges in our matching data structure to change. For each such changed edge, we carry out constant immediate processing, and also possibly create up to two "newly free" nodes. The crucial point is that if a newly free node is matched, then it can create at most one new newly free node, and so the total number of newly free nodes that have cascaded down the classes starting from a particular changed edge is at most 2.

For each weight class, we might need to consider up to $O(T(m,n)\varepsilon^{-1}\log C)$ newly free nodes from the classes above it, performing a constant time operation for each. Therefore the total time it takes to handle the entire merger is $O(\epsilon^{-2}\log^2(C)T(m,n))$. This process is sufficient to produce an output graph identical to one created by the static merge described above, since edges from higher classes are allowed to preempt ones from lower classes, just as if we'd greedily merged them in first. For the actual update time, we need to read in the weight of any inserted edge, and for this we pay an additional $O(\log N)$.

In Section 3 we provide the full details of the algorithm and the proofs of runtime, correctness and approximation guarantee.

1.1.2 Overview of improving the dependence on the weights

Here we provide an overview of our proof of Theorem 2 presented in Section 4.

A useful fact about weighted matching is that the maximum weight matching of the restriction of the graph to edges of weight at least a $2\epsilon/n$ fraction of the weight of the largest edge (so at least $2\epsilon N/n$, in our case) has weight at least $1/(1+\epsilon)$ times the true maximum weight matching, even if there are no edges other than the unique edge of weight N in this subgraph. The reason is simple: a matching has at most $n/2$ edges, and if none of these weigh more than $2\epsilon N/n$, the whole matching weighs only $(n/2)(2\epsilon N/n) = \epsilon N$, meaning the whole matching including the top edge weighs at most $(1+\epsilon)N$, as desired. Adding more edges above the threshold only makes the "best-only" matching a better approximation of the true matching.

It would be natural to want to apply this principle to fully dynamic weighted matching algorithms (like ours above) in order to reduce the dependence on the – potentially quite large – weight range to a dependence on $O(\epsilon^{-1}n)$. Unfortunately, N can potentially change in the fully dynamic setting: in particular, if we were relying on a single high-weight edge to carry our matching and that edge is deleted, we would have no approximation at all!

Our solution is to maintain enough copies of a dynamic maximum weight matching algorithm on a small range of weights, to guarantee that any two edges whose weights are within a factor of $2\epsilon/n$ of one another appear in the same data structure somewhere. In particular, the highest weight edge appears in the same data structure with any edge with weight within a $2\epsilon/n$ factor of the maximum weight. At the same time, we keep the weight ranges nearly disjoint so that only two of the data structures need to be altered by any update to the underlying graph.

To accomplish both goals, we take weight ranges of the form $((n\varepsilon^{-1})^i, (n\varepsilon^{-1})^{i+2}]$ for $\log_{n\varepsilon^{-1}} L \le i \le \log_{n\varepsilon^{-1}} N$ where L and N are the minimum and maximum edge weight in the data structure throughout its existence. The number of these ranges is now $O(\log_{n\varepsilon^{-1}} C)$ (where $C = N/L$), and we have guaranteed that the MWM in one of these classes is a $(1+\epsilon)\alpha$ approximation to the true MWM, and that the weight range in each class can be reduced to $n\varepsilon^{-1}$. However, we would like to have a persistent output matching that always contains a $(1+\epsilon)\alpha$ approximation of the maximum weight matching, that doesn't change dramatically in a single time step just because the highest weight edge was deleted from the graph.

To this end, we define and maintain a specially constructed output matching M that we call the *census matching*. The idea is as follows. M will contain a certain number of edges from the matchings in each weight class as follows. Consider some weight class $((n\varepsilon^{-1})^i, (n\varepsilon^{-1})^{i+2}]$ and suppose that the number of edges in its matching is n_i. Let $N_i = \sum_{j>i} n_j$ be the sum of cardinalities of matchings in classes above i. If $n_i > N_i$, we will have $n_i - N_i$ matching edges R_i from class i that the census matching M will consider. M is constructed by greedily merging R_i similar to our algorithm from the MWM to MCM reduction.

In particular, the census matching always contains all the edges of weight within $n\varepsilon^{-1}$ of the current maximum weight. Further, since the total number of edges in matchings that have any edges considered by the output matching doubles with each lower class that still has nonempty R_i, and since no matching has more than $n/2$ edges, the total number of classes with any edges considered is $O(\log n)$.

The output matching itself is relatively simple and behaves like the output matching of our first algorithm, except that it's drawing from these "representative sets" R_i, rather than from approximate maximum cardinality matchings. Fortunately, we can show that only two representative sets can change at a time, so it only takes $O(t(n, m, (n\varepsilon^{-1})^2, (n\varepsilon^{-1})^2) \log n)$ time to do the updates, where $t(n, m, N, C)$ is the update time of the underlying MWM algorithm, and thus the maximum number of edges that can be added or removed from a single MWM in a single time step.

We show that when an edge is to be updated, to figure out which pair of classes it belongs to, we only need to read $O(\log \log N)$ bits of its weight. To make everything work efficiently, we introduce a complete binary tree data structure (with the weight classes as leaves) that helps us maintain the representative sets R_i efficiently, though the tree is only conceptually complete, as we only add edges and nodes on paths from the root to leaves of non-empty weight classes to avoid wasting time when a new edge appears of much higher or lower weight than all previous edges. In particular, the $O(\log \log C)$ overhead in our running time is due to the tree having depth $O(\log \log C)$. The $\log \log N$ dependence is due to various pointers to positions in the bits of the weight. The details are in Section 4.

2 Merging matchings greedily

Here we prove a technical lemma used in both of our algorithms.

Let $S_1, \ldots, S_k$ be matchings in a graph G. For any edge (u, v), let $\ell(u, v) = \max\{i \mid (u, v) \in S_i\}$. Call a matching M a *Greedy Census* matching if for any edge $(u, v) \in S_i \setminus M$, there exists either (u, u') or $(v, v') \in M \cap S_j$, for some $j \geq i$. This property is equivalent to saying that M could have been constructed by greedily adding edges from each level from k down to 1, ensuring that the added edges are maximal within the current level before moving down. Thus, we have $\forall j : M \cap (\cup_{i>j} S_i)$ is maximal in $\cup_{i>j} S_i$

For an edge (u, v) let $L(u, v)$ be a decreasing set of indices $\{i_1, i_2, \ldots\}$ such that $(u, v) \in S_{i_j}$ for each j and $i_j > i_{j+1}$. In particular, $i_1 = \ell(u, v)$.

▶ **Lemma 4.** *Suppose we are given any collection of edge sets $In_1, In_2, \ldots, In_k$ and $Del_1, \ldots, Del_k$, the sets $L(u, v)$ for all (u, v) and a Greedy Census matching M of $S_1, \ldots, S_k$. Then one can insert all edges of In_j into S_j and delete all edges of Del_j from S_j and update M and all $L(u, v)$ so that M is a Greedy Census of the modified sets S_j, all in time*

$$\sum_{j=1}^{k} j \cdot \big(|In_j| + |Del_j|\big).$$

In our reduction from MWM to MCM, the sets S_j will correspond to the approximate maximum cardinality matchings of different weight classes, and in our algorithm for decreasing the dependence on the edge weight, they will correspond to the sets of representative edges of different weight classes.

Before we prove Lemma 4, let us introduce some notation.

For each $j \in [k]$, let NewFree(j), initially empty, be the set of newly free nodes created for level j. A newly free node u is any node that was covered by some edge in M, and then

became uncovered after a change to that matching. Specifically, u becomes newly free by the deletion of an edge (u, v) from M. This deletion could happen for one of two reasons: either because (u, v) was in $Del(j)$, meaning it was removed by an update to the underlying graph or some activity at a lower level of the algorithm, and is no longer in consideration for inclusion in M, or because $\ell(u, v) \leq j$ and (u, v) was deleted so that v can be matched via an edge of level greater than j; that is, we found another better edge for M to use in place of (u, v) and u was left uncovered.

Now the procedure is as follows. We set $\textsc{NewFree}(j) = \emptyset$ for all j. We iterate through all levels j from k down to 1. Fix a level j. Then, for each edge $(x, y) \in Del_j$, remove it from S_j, and remove j from $L(x, y)$. If now $\ell(x, y) < j$, and if $(x, y) \in M$, remove (x, y) from M and add x and y to $\textsc{NewFree}(j)$. For each $(x, y) \in In_j$, insert it into S_j, add j to $L(x, y)$. For each $(x, y) \in In_j$, in a second loop, check whether x and y are matched in M. Suppose that either x is not matched or x is matched to x' with $\ell(x, x') < j$. Suppose further that either y is not matched or it is matched to y' with $\ell(y, y') < j$. Then, remove (x, x') and (y, y') from M, add x' and y' to $\textsc{NewFree}(j)$ and insert (x, y) into M. If x or y are in $\textsc{NewFree}(j)$, remove them from $\textsc{NewFree}(j)$.

Now, for each $u \in \textsc{NewFree}(j)$, let v be its match in S_j (recall S_j is a matching). If v is matched to a node v' such that $\ell(v, v') \geq j$, then just move u to $\textsc{NewFree}(j-1)$ and move to the next u. Else if v is unmatched in M or if v is matched to some v' with $\ell(v, v') < j$, then remove (v, v') from M, add v' to $\textsc{NewFree}(j-1)$ and add (u, v) to M. This completes stage j.

Now we prove the following claims:

▶ **Claim 1.** *The runtime of the above algorithm is* $O(\sum_{j=1}^{k} j(|In_j| + |Del_j|))$.

Proof. Each deletion in Del_j can create at most 2 newly free nodes. Each insertion in In_j can cause the creation of at most 2 newly free nodes as well. If a newly free node is matched in some stage j, then it can create at most one new newly free node, and this one is put in $\textsc{NewFree}(j-1)$. Thus, the total cost of processing newly free nodes is $O(\sum_{j=1}^{k} j(|In_j| + |Del_j|))$. ◀

▶ **Claim 2.** *Let* $S'_i = S_i \cup In_i \setminus Del_i$. *$M$ is a greedy census matching of the updated matchings* $S'_1, \ldots, S'_k$.

Proof. Consider for contradiction an edge $(u, v) \in S_i \setminus M$ such that neither u nor v are matched with edges of level at least i. Before updating, one of the following was true, since M was a census matching: 1) $(u, v) \notin S_i$, 2) $(u, v) \in S_i \cap M$ or 3) (wlog) $(u, u') \in S_j \cap M$, for some $j > 1$. In case 1, (u, v) must have been in In_i, and, since neither u nor v are matched at a level above i, (u, v) would have been added to M, and then it's not true that $(u, v) \in S_i \setminus M$. In case 2, (u, v) left M with the update, but is still in S_i, so it wasn't in Del_i. This can only have happened because a higher level edge, wlog (u, u') was in $In_{\ell(u,u')}$, causing (u, v) to be removed from M. Since the In_i are processed in descending order, and the edges in $Del_{\ell(u,u')}$ get deleted before the edges from $In_{\ell(u,u')}$ get inserted, (u, u') will not be deleted by anything in this batch of updates, and so (u, u') is in M, and u is matched, violating the second part of the assertion. In case 3, the higher level edge, (u, u') must have been deleted in the update, either directly by being in Del_i or indirectly by having a higher level edge claim u'. In either case, u would be added to $\textsc{NewFree}(\ell(u', u))$, and it would either get matched to v, putting $(u, v) \in M$, or it would get matched to some (u, u'') of higher level than (u, v), both of which invalidate the assertion. ◀

3 A meta-algorithm for approximating MWM

Let $\varepsilon > 0$ be fixed. For every integer i, let E_i contain all edges of G that have weight $\geq (1+\varepsilon)^i$. Let D_i be a data structure that maintains an α-approximate MCM of E_i with update time $t(m,n)$.

Let ℓ be the smallest i such that $E_i \neq E_{i+1}$. During each stage of the dynamic algorithm we have a pointer to D_ℓ, for the current value of ℓ. Let M be the approximate MWM that we are maintaining. Let $\tilde{M}_i$ be the approximate MCM maintained by D_i, and let $M_i := M \cap \tilde{M}_i$. We will actually maintain M so that $M_i = M \cap E_i$.

We define the level $\ell(u,v)$ of edge (u,v) to be i such that $w(u,v) \in [(1+\varepsilon)^i, (1+\varepsilon)^{i+1})$.

We use Lemma 4 and its algorithm from Section 2 with $S_i = \tilde{M}_i$ to maintain the greedy census matching M. To do this, when a weighted edge (u,v) is inserted or deleted, we insert or delete it from all data structures D_j with $j \leq \ell(u,v)$. If $\ell(u,v) = \ell$ and $D_\ell \setminus D_{\ell+1}$ became empty, update ℓ. Now, after the D_j are updated, we figure out all the sets In_j and Del_j to feed into the data structure from Section 2.

We immediately obtain:

▶ **Lemma 5.** *M is a maximal matching in the graph $\cup_i \tilde{M}_i$.*

▶ **Lemma 6.** *The update time is $O(t(m,n)(\log_{1+\varepsilon} N/L)^2)$.*

Proof. Let $\mu = \log_{1+\varepsilon} N/L$. $|Del_j|$ and $|In_j|$ for each j are at most $O(t(m,n))$ since the (amortized/expected/worst case) number of deletions or insertions performed by each D_j is $\leq t(m,n)$. By the proofs in Section 2, the running time should be asymptotically $\sum_j j \cdot |In_j| + |Del_j| \leq t(m,n) \sum_{j=1}^{\mu} j \leq O(\mu^2 t(m,n))$. ◀

The proof of the Lemma below directly follows from Claim 2 from Section 2.

▶ **Lemma 7.** *For every j, $M \cap E_j$ is maximal matching in the graph $\cup_{i \geq j} \tilde{M}_i$*

Now we can prove the approximation guarantee part of our result.

▶ **Lemma 8.** *If each $\tilde{M}_j$ is an α-approximate MCM, then M is a $2\alpha(1+\varepsilon)$-approximate MWM.*

Proof. Consider a fixed optimal MWM M^*, and let m_i be the cardinality of the MCM on edges of level i, and note that $|M^* \cap E_i| \leq m_i$. Clearly $\alpha|\tilde{M}_i| \geq m_i \geq |M^* \cap E_i|$. Further, $2|M \cap E_i| \geq |\tilde{M}_i|$ by Lemma 7, since $M \cap E_i$ is a maximal matching on a superset of the edges in $\tilde{M}_i$, meaning $|M \cap E_i| \geq \frac{1}{2\alpha}|M^* \cap E_i|$. This means that for each i, every 2α edges of M^* can be assigned to a single "babysitter" edge of M of equal or higher level – this is perhaps easier to see by subtracting out all of the already assigned "pairs" from higher levels, giving $|M \cap E_i| - \frac{1}{2\alpha}|M^* \cap E_{i+1}| \geq \frac{1}{2\alpha}(|M^* \cap E_i| - |M^* \cap E_{i+1}|)$. Since the babysitter edge e is of at least as high a level as all of its charges, $(1+\varepsilon)w(e) \geq w(e^*)$, and, in general, $2(1+\varepsilon)\alpha w(M \cap E_i) \geq w(M^* \cap E_i)$, which, taking $i = L$, proves the lemma. ◀

4 A Meta-meta-algorithm for approximating MWM

4.1 The Census Matching

We maintain a matching using the process from Section 2, partitioning the weight range, running a matching algorithm on each partition, and merging the results, just like in Section 3. However, we keep weighted (rather than unweighted) matchings in each class, and we only consider some of the edges from a few of these classes, rather than merging all of the edges.

In particular, we "semi-partition" the weight range, such that every value within that range falls into exactly two of our semi-partition intervals. Each interval is of the form $((n/\varepsilon)^i, (n/\varepsilon)^{i+2}]$, for $i \in \mathbb{Z}$ such that $\log(w_o) - 1 \leq i \leq \log(w^*) - 1$, where w_o and w^* are the highest and lowest weights that have appeared on any edge in the matching, even if that edge was later deleted. For each of these semi-partitions, we maintain a maximum weight matching algorithm on the subgraph defined by the edges in the underlying graph whose weights fall within that interval. We denote by W_i the approximate MWM maintained on the interval $((n/\varepsilon)^i, (n/\varepsilon)^{i+2}]$.

To merge the W_is together, we maintain a single census matching C, which uses the algorithm from Section 2 to combine a subset of the edges from $O(\log n)$ specific W_is, as determined by a binary tree data structure that we will call the "responsibility tree", T. We will describe this below.

As described in the introduction, we strive to select from each W_i a number of representative edges. Let n_i be the cardinality of W_i, and let $N_i = \sum_{j>i} n_j$ be the total sum of the cardinalities of the matchings in classes above i. If $n_i > N_i$, we will have $n_i - N_i$ matching edges R_i from class i that the census matching M will consider. The tree data structure T will facilitate maintaining the cardinalities that the R_is need to have.

4.2 The Responsibility Tree

We build a near-complete binary tree with leaves corresponding to the W_is to efficiently determine which edges should be sent to the census matching to ensure the desired properties. The W_is are arranged in descending order from left to right, with the leftmost leaf corresponding to $W_{\log_{n/\varepsilon}(w^*)}$ and the rightmost to $W_{\log_{n/\varepsilon}(w_o)}$. The leaves track the number of edges in the matchings and the number of those edges which are represented, and the internal nodes have attributes which depend on the attributes of their children. Whether or not each W_i has the correct number of represented edges can be determined just by looking at the attributes of the root, and if those attributes are wrong, the incorrectly represented classes/leaves can each be tracked down in a single traversal from root to leaf.

Every node in the tree has three attributes: mass, high, and low. The mass attribute is a running tally of the total number of edges among matchings in the classes associated with the leaves of this subtree. The "high" and "low" attributes validate the number of representatives that classes in this subtree have: if the "high" indicator $h(v)$ on any node v is negative, some class in that node's subtree $t(v)$ has too *many* representatives, and if the "low" indicator $l(v)$ on any node v in the left-most branch (the "spine") is positive, then some class in that node's subtree $t(v)$ has too *few* representatives.

We refer to edges that are in the selection pool for the census matching as "representatives," and we record the number of representatives for each class as $R(v)$.

We refer to the high indicator for a vertex v as $h(v)$, its left child as $v.l$, and its right child as $v.r$.

$$h(v) := \begin{cases} \min(h(v.l), h(v.r) - m(v.l)) & \text{if } v \text{ is not a leaf} \\ \infty & \text{if } leaf(v) \wedge R(v) = 0 \\ m(v) - R(v) & \text{if } leaf(v) \wedge R(v) > 0 \end{cases}$$

Similarly for the low indicator, $\ell(v)$,

$$\ell(v) := \begin{cases} m(v) - R(v) & \text{if } v \text{ is a leaf} \\ \max(h(v.l), h(v.r) - m(v.l)) & \text{otherwise} \end{cases}$$

Finally, mass, $m(v)$, is simply

$$m(v) := \begin{cases} \#\text{ edges in matching} & \text{if } v \text{ is a leaf} \\ m(v.l) + m(v.r) & \text{otherwise} \end{cases}$$

4.2.1 Tree Properties

▶ **Lemma 9.** *If the root r has $h(r) = l(r) = 0$, then for every class ℓ_i, $m(\ell_i) = R(\ell_i) + \sum_{j>i} m(\ell_j)$ if $R(\ell_i)$ is positive, and $m(\ell_i) \leq \sum_{j>i} m(\ell_j)$ otherwise. That is, if the root's high and low indicators are both 0, then no class has too many or too few representatives.*

We will prove the lemma using two claims below.

▶ **Claim 3** (No Class has Too Many Representatives.)**.** *For any node v,*

$$h(v) = \min_{\{a_i \in t(v) | R(a_i) \neq 0\}} \left(m(a_i) - R(a_i) - \sum_{j>i \in t(v)} m(a_j) \right).$$

Proof. By induction. For a leaf a, the subtree $t(a)$ is trivial, and this is the definition of $h(a)$. For an internal node v, $h(v) = \min(h(v.l), h(v.r) - m(v.l))$. $v.l$ has no higher classes to account for in $t(v)$, and subtracting $m(v.l)$ from $h(v.r)$ subtracts the masses of all previously unaccounted for classes, meaning both children fit the conditions, and $h(v)$ is simply the minimum of these, as desired. ◀

Consider a class a_i such that its number of represented edges is greater than it should be, i.e., the total number of edges in classes above it $\sum_{j>i}(m(a_j))$, plus the number of its representatives $R(a_i)$, is greater than its own mass $m(a_i)$. Then by Claim 3, $h(r)$ will be negative.

Thus, if $h(r) = 0$, there is no class with too many representatives.

▶ **Claim 4** (No Class has Too Few Representatives.)**.** *For any node v,*

$$\ell(v) = \max_{a_i \in t(v)} \left(m(a_i) - R(a_i) - \sum_{j>i \in t(v)} m(a_j) \right).$$

Proof. By induction, symmetrical to proof of Claim 3. ◀

Consider a class a_i such that its number of represented edges is less than it should be, i.e., $m(a_i) - R(a_i) - \sum_{j>i}(m(a_j)) > 0$. Then by Claim 4, $\ell(r) > 0$.

Thus, if $\ell(r) = 0$, no node has too few representatives.

4.3 The Census Matching is Nearly as Good as The Underlying MWM

We define the "Census matching" to be the resultant matching from combining the representative edges from each class in a way that mimics a static greedy merge, that is, using the algorithm of Section 2.

4.3.1 The Best is Good Enough

▶ **Lemma 10.** *There exists a matching, M_B, such that 1) the ratio between the lowest and highest weight edges in M_B is at most $\epsilon^{-1}n/2$ and 2) $(1+\epsilon)w(M_B) \geq w(M^*)$*

Proof. Consider the lowest interval of width $\epsilon^{-1}n/2$ such that there are no edges above that interval; thus, the interval will contains the highest weight edge(s) in the underlying graph at the top, as well as all edges that weigh at least $2\epsilon/n$ times as much as those edges. We call this interval $B = [2\epsilon/nw^*, w^*]$, where w^* is the highest weight of any edge in the underlying graph.

Consider the optimal matching on edges with weights that fall within B, $\textsc{Opt}_b$. Clearly, its weight is at least w^*, since the trivial matching formed by taking just the single highest weight edge has that weight, and is a matching on edges in B. Now consider the amount by which the true optimum matching, $\textsc{Opt}$, exceeds the weight of $\textsc{Opt}_b$. Any gains the true optimum makes must be from the inclusion of edges outside of (hence, below) B, each of which weighs at most $2\epsilon w^*/n$. Further, the optimum matching can only include $n/2$ such edges (because it is a matching), meaning

$$\begin{aligned} w(\textsc{Opt}) - w(\textsc{Opt}_b) &\leq \frac{n}{2}\frac{2\epsilon w^*}{n} && (1)\\ &= \epsilon w^* && (2)\\ &\leq \epsilon w(\textsc{Opt}_b) && (3) \end{aligned}$$

which tells us that

$$w(\textsc{Opt}) \leq (1+\epsilon)w(\textsc{Opt}_b) \quad (4)$$

◀

So if we have a maximum weight matching algorithm that gives an α-approximation, and run it on just edges with weights in B, we get a $(1+\epsilon)\alpha$-approximation to the overall maximum weight matching. Moreover, if we run it on edges from some interval containing B, we get the same guarantees.

▶ **Corollary 11.** *For any graph G, there exists an interval $I = [(\epsilon^{-1}n/2)^i, (\epsilon^{-1}n/2)^{i+2})]$ such that for any MWM algorithm A, $(1+\epsilon)w(A(G_I)) \geq w(A(G))$, where G_I is the restriction of G to edges with weights in I.*

4.3.2 The Census Displays the Best

▶ **Lemma 12.** $w(M) \geq w(M_B)$

Proof. Since we have a copy of our MWM algorithm running for each non-empty interval of the form $((\epsilon^{-1}n)^i, (\epsilon^{-1}n)^{i+2}]$, we have a MWM covering a superset of every interval of "width" $\epsilon^{-1}n/2$, and, in particular, B. By Lemma 1, the top non-empty class (which contains B) has all of its edges represented, and by the fact that the census matching prioritizes higher weight edges, all of these representatives will, in fact, be included in the final census matching. This means that the census matching is a superset of MWM_B, meaning its weight is at least $(1+\epsilon)\alpha w(\textsc{Opt})$. ◀

4.4 Maintaining the Census is Fast

▶ **Lemma 13.** *Given an underlying MWM running in time $T(n, m, C, N)$, the census matching can be maintained in $O(\log^2(n)T(n, m, (n/\varepsilon)^2, (n/\varepsilon)^2))$ time.*

Proof. By Lemma 9, any class with 1 or more representatives has more edges in its underlying matching than every class above it. Clearly, this means that each class with any representatives has twice as many edges in its matching than the last. Combined with the fact that no matching can have more than $n/2$ matchings, this means there are at most $O(\log n)$ represented classes, and consequently only $O(\log n)$ levels in the census matching. Then the proof follows from the Lemma in Section 2.

Within each of the classes, an instance of the underlying MWM is being run with weights between 1 and $(n/\varepsilon)^2$, and so the maximum number of edges that can change in any given matching (and thus the cardinality of Del_i and In_i) is bounded above by the running time of the MWM on that interval, i.e. $T(n, m, (n/\varepsilon)^2, (n/\varepsilon)^2)$ ◀

4.5 Maintaining the Tree is Fast

▶ **Lemma 14.** *Updating the tree after an insertion or deletion takes $O(\log n \log\log C)$ time*

Proof. Since the Responsibility Tree is a near-complete binary tree with $O(\log C)$ leaves, the length of the path from a leaf to the root is $O(\log\log C)$, meaning only $O(\log\log C)$ nodes need to be changed for each leaf whose underlying matching had a change in cardinality or represenatives. Since any edge is within the purview of only two W_is, only two classes can have a change in the cardinality of their underlying matchings.

By the proof of Lemma 13, $O(\log n)$ classes have a non-zero number of representatives before and after the update, meaning that for all but $O(\log n)$ classes, the number of representatives was 0 before and after the change. So the total number of classes which had a change in number of representatives or cardinality is at most $O(\log n)$, which, combined with the fact that updating each one takes $O(\log\log C)$ time, gives us the desired time complexity. ◀

4.6 Numerical Considerations (reading an update is fast)

We assume that updates arrive with weights w that are organized like standard floating point numbers: a significand s, a base b, and an exponent x such that $1 \leq s < b$ and $sb^x = w$, along with pointers to where in the number these segments begin and end. Note that s will be $O(\log w)$ bits long, b will be a constant number of bits, and x will be $O(\log\log w)$ bits.

We want two pieces of information from this update: the pair of W_is that the edge should be processed by, and enough information about the edge's weight to maintain those W_is without too much error. The first piece of information is easy to approximate from reading just the exponent and the base, since the significand only changes the weight of w by a factor of at most b. Then we can send the edges to class i and $i+1$ for $i = x/\log_b(n/\varepsilon)$. The only concern then is if $x/\log_b(n/\varepsilon) < i+1$ but $sx/\log_b(n/\varepsilon) = \log_{n/\varepsilon} w \geq i+1$, in which case the edge "belonged" in classes $i+1$ and $i+2$ but was sent to i and $i+1$ instead. Fortunately, if w is on the border in this way, and $i+2$ is the highest non-empty class, w would be on one of the lowest weight edges in that class, meaning that its weight is at most $\varepsilon b/n$ times the weight of the largest edge in the matching, and the error incurred is only $O(\varepsilon)$ times that weight, even if a full $n/2$ edges of this type are missed, and so the error is subsumed into the rounding error.

The second piece of information does require us to read the significand, but fortunately only very few bits of it. Since all of the edges within class i have weights between $(n/\varepsilon)^i$ and $(n/\varepsilon)^{i+2}$, and because misreporting the lower order bits of the weights only causes multiplicative $(1-\varepsilon/n)$ error in each edge weight, for a total error of ε from all edges in the matching, we can divide the weights of the edges by the lower bound of their classes, resulting in needing to send weights that can be described in only $O(\log(n/\varepsilon))$ bits. Further, since we can just truncate off the trailing bits of the significand rather than performing a true division, we only need to read the first $O(\log(n/\varepsilon))$ bits of the significand to determine which values to send.

References

1. Amir Abboud and Virginia Vassilevska Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 434–443, 2014.
2. A. Anand, S. Baswana, M. Gupta, and S. Sen. Maintaining approximate maximum weighted matching in fully dynamic graphs. In *FSTTCS*, pages 257–266, 2012.
3. S. Baswana, M. Gupta, and S. Sen. Fully dynamic maximal matching in $O(logn)$ update time. In *FOCS*, pages 383–392, 2011.
4. Aaron Bernstein and Cliff Stein. Fully dynamic matching in bipartite graphs. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 167–179, 2015.
5. Aaron Bernstein and Cliff Stein. Faster fully dynamic matchings with small approximation ratios. In *In Proc. SODA*, page to appear, 2016.
6. Sayan Bhattacharya, Monika Henzinger, and Giuseppe F. Italiano. Deterministic fully dynamic data structures for vertex cover and matching. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 785–804, 2015.
7. Sayan Bhattacharya, Monika Henzinger, and Danupon Nanangkai. New deterministic approximation algorithms for fully dynamic matching. In *Proc. STOC*, page to appear, 2016.
8. Michael Crouch and Daniel Stubbs. Improved streaming algorithms for weighted matching, via unweighted matching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014, Barcelona, Spain*, pages 96–104, 2014.
9. H. Gabow. A scaling algorithm for weighted matching on general graphs. In *Prof. FOCS*, pages 90–100, 1985.
10. H. N. Gabow and R. E. Tarjan. Faster scaling algorithms for general graph-matching problems. *J. ACM*, 38(4):815–853, 1991.
11. M. Gupta and R. Peng. Fully dynamic $(1+\varepsilon)$-approximate matchings. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 548–557, 2013.
12. N. J. A. Harvey. Algebraic structures and algorithms for matching and matroid problems. In *Proc. FOCS*, volume 47, pages 531–542, 2006.
13. J. Hopcroft and R. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.
14. Jonathan A. Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on*

Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014, pages 217–226, 2014.

15 H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

16 François Le Gall. Powers of tensors and fast matrix multiplication. In *International Symposium on Symbolic and Algebraic Computation, ISSAC '14, Kobe, Japan, July 23-25, 2014*, pages 296–303, 2014.

17 A. Madry. Navigating central path with electrical flows: from flows to matchings, and back. In *Proc. FOCS*, 2013.

18 Silvio Micali and Vijay V. Vazirani. An o(sqrt(|v|) |e|) algorithm for finding maximum matching in general graphs. In *21st Annual Symposium on Foundations of Computer Science, Syracuse, New York, USA, 13-15 October 1980*, pages 17–27, 1980.

19 M. Mucha and P. Sankowski. Maximum matchings via gaussian elimination. In *Proc. FOCS*, volume 45, pages 248–255, 2004.

20 K. Mulmuley, U. V. Vazirani, and V. V. Vazirani. Matching is as easy as matrix inversion. In *Proc. STOC*, volume 19, pages 345–354, 1987.

21 O. Neiman and S. Solomon. Simple deterministic algorithms for fully dynamic maximal matching. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 745–754, 2013.

22 Krzysztof Onak and Ronitt Rubinfeld. Maintaining a large matching and a small vertex cover. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 457–464, 2010.

23 M. O. Rabin and V. V. Vazirani. Maximum matchings in general graphs through randomization. *J. Algorithms*, 10(4):557–567, 1989.

24 P. Sankowski. Faster dynamic matchings and vertex connectivity. In *Proc. SODA*, pages 118–126, 2007.

25 P. Sankowski. Maximum weight bipartite matching in matrix multiplication time. *Theor. Comput. Sci.*, 410(44):4480–4488, 2009.

26 V. Vassilevska Williams. Multiplying matrices faster than Coppersmith-Winograd. In *Proc. STOC*, pages 887–898, 2012.

SOS Is Not Obviously Automatizable, Even Approximately*

Ryan O'Donnell

Computer Science Department, Carnegie Mellon University, Pittsburgh, USA
odonnell@cs.cmu.edu

Abstract

Suppose we want to minimize a polynomial $p(x) = p(x_1, \ldots, x_n)$, subject to some polynomial constraints $q_1(x), \ldots, q_m(x) \geq 0$, using the Sum-of-Squares (SOS) SDP hierarachy. Assume we are in the "explicitly bounded" ("Archimedean") case where the constraints include $x_i^2 \leq 1$ for all $1 \leq i \leq n$. It is often stated that the degree-d version of the SOS hierarchy can be solved, to high accuracy, in time $n^{O(d)}$. Indeed, I myself have stated this in several previous works.

The point of this note is to state (or remind the reader) that this is not obviously true. The difficulty comes not from the "r" in the Ellipsoid Algorithm, but from the "R"; a priori, we only know an exponential upper bound on the number of bits needed to write down the SOS solution. An explicit example is given of a degree-2 SOS program illustrating the difficulty.

1998 ACM Subject Classification G.1.6 Optimization

Keywords and phrases Sum-of-Squares, semidefinite programming

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.59

1 Introduction

Suppose you want to approximately minimize a real polynomial $p(x) = p(x_1, \ldots, x_n)$ over the set $K = \{x \in \mathbb{R}^n : q_1(x) \geq 0, \ldots, q_m(x) \geq 0\}$, where $q_1, \ldots, q_m$ are real polynomials. All of the examples I'll consider will be quite simple: m will be at most $O(n)$; and, the polynomials $p, q_1, \ldots, q_m$ will be of degree at most 2 and will have small integer coefficients (magnitude at most poly(n), say; often at most 2). A good example to keep in mind arises from the "Balanced Separator" problem in combinatorial optimization. There, you're given an n-vertex graph $G = (V, E)$ and the goal is to partition its vertices into two parts, neither of size more than $\frac{2}{3}n$, such that the number of edges crossing between the parts is minimized. Introducing a variable x_i for each vertex, this is equivalent to solving

$$\min \sum_{\{i,j\} \in E} \tfrac{1}{4}(x_i - x_j)^2 \quad \text{subject to} \quad \{x_i^2 = 1 \ \forall i, \ -\tfrac{1}{3}n \leq x_1 + \cdots + x_n \leq \tfrac{1}{3}n\}.$$

Here $x_i^2 = 1$ can be treated as the two inequalities $1 - x_i^2 \geq 0$, $-1 + x_i^2 \geq 0$. Another good example arises from the "Maximum Independent Set" problem on G:

$$\max \sum_{i=1}^{n} x_i \quad \text{subject to} \quad \{x_i^2 = x_i \ \forall i, \ x_i x_j = 0 \ \forall \{i,j\} \in E\}.$$

A powerful technique for trying to certify that the minimum is at least $\theta \in \mathbb{R}$ is to find a formal polynomial identity of the form

$$p(x) - \theta = u_0(x) + u_1(x)q_1(x) + \cdots + u_m(x)q_m(x), \tag{1}$$

* This work was partially supported by NSF grant CCF-1618679.

© Ryan O'Donnell;
licensed under Creative Commons License CC-BY

where each $u_j(x)$ is SOS; i.e., a sum of squares of polynomials. We will refer to this as "SOS-proving" or "SOS-certifying" the statement "$p(x) \geq \theta$". A variation of this technique ("SOS-refutation") is to take $q_{m+1}(x) = (\theta - \epsilon) - p(x) \geq 0$ as an additional constraint, and then try to SOS-prove the statement "$-1 \geq 0$". It's easy to check that we can do this for every $\epsilon > 0$ provided that "$p(x) \geq \theta$" is SOS-provable. So if we aren't concerned with very small additive errors — and I won't be, in this note — the refutation technique is fundamentally stronger (see, e.g., [22] for further discussion). In any case, I'll use the "SOS-certifying" terminology in the rest of the note, since SOS-refutation is just a special case with one extra constraint.

Suppose we now bound the degree of the $u_j(x)$'s, insisting that $\deg(u_0), \deg(u_1 q_1), \cdots \leq d$. Then the question of whether the certifying $u_j(x)$'s exist is equivalent to the feasibility of a certain semidefinite program (SDP). This is the "degree-d SOS relaxation", pioneered by Shor [28], Nesterov [20], Grigoriev and Vorobjov [9], Lasserre [16, 17] and Parrilo [23]. See, e.g., [18, 3] for many more details.

Under the simple assumptions I mentioned (namely, $m \leq O(n)$, p and q_j's having small coefficients and degree at most 2), the degree-d SOS SDP for (1) can be written down using $N = n^{O(d)}$ bits. It is then quite commonly stated that feasibility can be tested in $\text{poly}(N)$ time, using, say the Ellipsoid Algorithm [15, 11]. This is sometimes referred to as the SOS proof system being "automatizable". Unfortunately, I will now explain why it's not clear whether this is truly the case.

Approximation, and the explicitly bounded case. I should emphasize that I am *not* worried about very small additive errors; i.e., the difference between testing feasibility and near-feasibility. Indeed, most often the caveat is correctly added that semidefinite programming only tests feasibility up to a very small additive error. This caveat is related to the fact that the Ellipsoid Algorithm has a technical requirement, that if the SDP is feasible then it contains a feasible ball of some small radius $r = 2^{-\text{poly}(N)} > 0$. Actually, to talk about additive error only makes sense if there is some notion of "scaling". To continue keeping things simple, I'll henceforth assume that the variables are intended to be in the range $[-1, 1]$; i.e., that K always includes the constraints $x_i^2 \leq 1$ for $1 \leq i \leq n$. (It would actually be fine if we even just had $x_i^2 \leq 2^{\text{poly}(N)}$ for all i.) This is sometimes called the "explicitly bounded" or "Archimedean" case, and it's also known to imply that the SDP has no duality gap [13].

With this issue discussed, let's now again pose the question:

Question. *Suppose there is a degree-d SOS proof that $p(x) \geq \theta$ subject to constraints $x_1^2, \ldots, x_n^2 \leq 1$ and $q_1(x), \ldots, q_m(x) \geq 0$, of the form (1). Is there a $\text{poly}(N)$-time algorithm (presumably, a version of the Ellipsoid Algorithm) that finds SOS polynomials $u_0(x), \ldots, u_m(x)$ certifying $p(x) \geq \theta - o_N(1)$?*

In a joint work with Yuan Zhou [22, Footnote 2], I wrote that the answer is "yes".[1] However I now see that my reasoning was incomplete, and that the answer is unclear. In fact, I would now guess that the answer is probably "no". Although it's true that the technical "r" parameter in the Ellipsoid Algorithm does not cause real problems in the explicitly bounded case, there is another technical parameter, "R" — and it *does* seem to cause real problems. The Ellipsoid Algorithm is only guaranteed to work correctly in $\text{poly}(N)$ time if the SDP's

[1] Sorry for talking you into that footnote, Yuan.

feasible region (should it exist) intersects a ball of radius $R = 2^{\mathrm{poly}(N)}$. In other words, algorithmically speaking it's not enough for an SOS proof to exist; we also need one to exist in which all the SOS polynomials can be written down with $\mathrm{poly}(N)$ bits. However, in the next section I'll show a simple, explicitly bounded example where an inequality is SOS-provable, but any approximate SOS proof requires integers of size roughly 2^{2^n}. This example is based on the well-known fact (attributed to J. Ramana in [1] and to Khachiyan in [25]) that there are SDPs with n variables and $O(n)$ constraints that are feasible, yet for which every feasible solution requires exponential bit-complexity.

In fact, as pointed out to me by Pablo Parrilo, *every* SDP-feasibility problem can be viewed as an SOS-feasibility problem modulo an ideal; thus, if we ignore the insistence on $x_i^2 \le 1$ constraints, the above **Question** is tantamount to simply asking if the Semidefinite Feasibility Problem (SDFP) is in P. This is a well-known open question; see [25, 24, 30]. The best current upper bound known is PSPACE, by reduction to the existential theory of the reals.[2]

2 SOS-provable, but only with huge coefficients

Let's say we have $2n$ indeterminates $x_1, x_2, \dots, x_n, y_1, \dots, y_n$, and the following constraints.

$$\begin{array}{llll} 2x_1y_1 = y_1, & 2x_2y_2 = y_2, & 2x_3y_3 = y_3, & 2x_ny_n = y_n, \\ x_1^2 = x_1, & x_2^2 = x_2, & x_3^2 = x_3, \quad \cdots & x_n^2 = x_n, \\ y_1^2 = y_2, & y_2^2 = y_3, & y_3^2 = y_4, & y_n^2 = 0. \end{array} \tag{K}$$

(These should be read column-wise. Notice the very last constraint, $y_n^2 = 0$, breaks the pattern.)

At first, I won't include the constraints $x_i^2 \le 1$, $y_i^2 \le 1$; we'll analyze their inclusion later.

We wish to know whether

$$p_n(x, y) = x_1 + x_2 + x_3 + \cdots + x_n - 2y_1$$

is nonnegative subject to these constraints. It's easy for the human mathematician to see the answer is "yes", because "solving" the constraints shows that they are equivalent to $x_1, \dots, x_n \in \{0, 1\}$ and $y_1 = \cdots = y_n = 0$; hence the minimum of $p_n(x)$ is 0. However SOS algorithms do not first try to "solve" or "simplify" the constraints.[3] So we have to see what happens when SOS algorithms are run "generically" on this input.

For simplicity, let's consider the degree-2 SOS algorithm. In this case, whether we consider the constraints as equalities or two-inequalities amounts to the same thing: we get to multiply

[2] Even the special case of deciding whether a given rational multivariate polynomial is SOS is not known to be in P or even in NP. I do not know if the "R" problem is relevant here, but the "r" problem certainly is; according to Scheiderer [26] there are rational polynomials such as $x^4 + xy^3 + y^4 - 3x^2yz - 4xy^2z + 2x^2z^2 + xz^3 + yz^3 + z^4$ that are SOS but don't have a rational SOS representation. However in this work I am less concerned with this kind of example, because I would like to consider the "bounded case" and allow approximation.

[3] Otherwise, the well-known SOS lower bounds for "Knapsack" [8] and "kXOR" [7, 27] would be invalid. In particular, applying a Gröbner basis algorithm to the constraints is not a good idea in general, since it has exponential complexity even for zero-dimensional ideals [12]. For example, the size of the Gröbner basis for the very simple "Max-Bisection" ideal, $\{x_1^2 = \cdots = x_{2n}^2 = 1, x_1 + \cdots + x_{2n} = 0\}$, is $\widetilde{\Theta}(2^n)$.

them by nonnegative reals. Thus the question becomes:

$$p_n(x,y) = \big[\text{degree-2 SOS}\big] \bmod (\text{K})? \tag{2}$$

where the "mod" refers to adding linear multiples of the constraint equations — i.e., adding $a(2x_1y_1-y_1)+b(x_1^2-x_1)+c(y_1^2-y_2)+d(2x_2y_2-y_2)+\cdots$ for some real constants $a,b,c,d,\ldots$. The answer to this question is *also* "yes":

$$p_n(x,y) = (x_1-2y_1)^2+(x_2-4y_2)^2+(x_3-16y_3)^2+(x_4-256y_4)^2+\cdots+(x_n-2^{2^{n-1}}y_n)^2 \bmod (\text{K}).$$

However, it turns out that *every* way of expressing $p_n(x,y)$ as in (2) has exponential-in-n bit-complexity. This shows that no matter how exactly we formulate the SOS problem as an SDP (e.g., whether we look for homogeneous or non-homogeneous sums of squares, whether we explicitly introduce variables $a,b,c,d,\ldots$ to multiply against the constraints or instead work "mod the ideal", etc.), no generic polynomial-time SDP-solving algorithm will find a degree-2 SOS proof of $p_n(x,y) \geq 0$.[4]

Before proving this, two comments: First, this example and its proof are nothing more than a slight rearrangement of the standard example of a feasible SDP whose only feasible solutions are doubly exponential. I'm only putting an SOS spin on it. Second, this argument doesn't really give a negative example for the **Question** from Section 1, because it's conceivable that there is a degree-2 SOS proof with polynomial bit-complexity of "$p_n(x,y) \geq -\epsilon_n$", where $\epsilon_n = o_n(1)$. In Subsections 2.1, 2.2, I'll show that even this is impossible, even when the constraints $x_i^2 \leq 1$, $y_i^2 \leq 1$ are added.

So let's suppose we have an SOS representation of $p_n(x,y)$ as in (2):

$$x_1 + x_2 + \cdots + x_n - 2y_1 = \sum_j \ell_j(x,y)^2 \bmod (\text{K}), \tag{3}$$

where the ℓ_j's denote linear polynomials. In fact, the ℓ_j's must be homogeneous of degree 1. The reason is that if we set all x_i's and y_i's to 0 in (3), the LHS becomes 0 and the RHS becomes the sum of the squares of the constant coefficients of the ℓ_j's. Hence all these constant coefficients must be 0.

Next, let us express each ℓ_j as $\sum_{k=1}^n \ell_{jk}$, where each ℓ_{jk} is of the form $a_{jk}x_k + b_{jk}y_k$. It would of course be incorrect to say that $\ell_j^2 = \sum_{k=1}^n \ell_{jk}^2$ — to neglect the cross-terms is the so-called "freshman's dream". Notice, though, that any nonzero cross-term contains a monomial of the form $x_kx_{k'}$, $x_ky_{k'}$, or $y_ky_{k'}$ ($k \neq k'$), and no such monomial appears on the left in (3). Furthermore, such monomials are not affected by the "mod (K)", and thus they must be canceled via cross-terms arising from other squares $\ell_{j'}^2$ in the sum. Hence following the "freshman's dream" in $\sum_j \ell_j^2$ actually gives the same identity in (3). In other words, we may assume without loss of generality that (3) is of the form

$$\sum_{i=1}^n \sum_j (a_{ij}x_i + b_{ij}y_i)^2 = \sum_{i=1}^n (A_i^2x_i^2 + 2M_ix_iy_i + B_i^2y_i^2), \tag{4}$$

where $A_i = \sqrt{\sum_j a_{ij}^2}$, $B_i = \sqrt{\sum_j b_{ij}^2}$, and $M_i = \sum_j a_{ij}b_{ij}$. Cauchy–Schwarz implies

$$|M_i| = \sum_j a_{ij}b_{ij} \leq A_iB_i. \tag{5}$$

[4] Note that it doesn't matter whether we ask the algorithm to find a PSD matrix representing the SOS polynomial, or the actual sums of squares. Since Cholesky (LDL) decomposition can be done in polynomial time (see Section 4), if there were a rational PSD matrix of polynomial bit-complexity representing the SOS polynomial, we could extract from it an explicit rational sum-of-squares representation with polynomial bit-complexity.

For (4) to equal the LHS of (3) mod (K), we'll need to use all of the equality constraints, thereby obtaining

$$\sum_{i=1}^{n}(A_i^2 x_i + M_i y_i + B_i^2 y_{i+1}),$$

with y_{n+1} denoting 0. Equating coefficients with LHS(3), we deduce

$$A_i = 1 \ \forall i, \quad M_1 = -2, \quad M_{i+1} = -B_i^2 \ \forall 1 < i < n.$$

Combining this with (5), we get $B_i \geq |M_i|$ for all i, and hence

$$B_1 \geq 2, \quad B_{i+1} \geq B_i^2 \ \forall 1 < i < n.$$

Thus $B_n \geq 2^{2^{n-1}}$; i.e., the sum of the squares of the coefficients on y_n in any representation (3) is at least 2^{2^n}. So indeed any solution to (2) has exponential bit-complexity.

2.1 Even approximately

I'll now show that even getting a degree-2 SOS proof of $p_n(x, y) \geq -o_n(1)$ is impossible without exponential bit-complexity. So suppose we have

$$x_1 + x_2 + \cdots + x_n - 2y_1 + \epsilon = \sum_j \ell_j(x, y)^2 \text{ mod (K)}, \tag{6}$$

where $\epsilon \leq .01$, say. Now we can't deduce that the ℓ_j's are homogeneous, but the reasoning concerning the "freshman's dream" still holds. So the SOS part must be of the form

$$\sum_{i=1}^{n}\sum_j (a_{ij}x_i + b_{ij}y_i + c_{ij})^2 = \sum_{i=1}^{n}(A_i^2 x_i^2 + 2M_i x_i y_i + B_i^2 y_i^2 + 2U_i x_i + 2V_i y_i + C_i^2),$$

where we're now introducing the notation $U_i = \sum_j a_{ij}c_{ij}$, $V_i = \sum_j b_{ij}c_{ij}$, and $C_i = \sqrt{\sum_j c_{ij}^2}$. Cauchy–Schwarz still implies (5), and also

$$|U_i| \leq A_i C_i, \quad |V_i| \leq B_i C_i. \tag{7}$$

Again, equating coefficients and reducing mod (K) yields

$$\epsilon = \sum_i C_i^2, \quad A_i^2 + 2U_i = 1 \ \forall i, \quad M_1 + 2V_1 = -2, \quad M_{i+1} + 2V_{i+1} = -B_i^2 \ \forall 1 < i < n. \tag{8}$$

As $\epsilon \leq .01$, the first equation implies $C_i \leq .1$ for all i. Thus (7) implies $|U_i| \leq .1A_i$, $|V_i| \leq .1B_i$. Substituting these into the above yields the following:

$$A_i^2 - .2A_i \leq 1 \implies A_i \leq 1.2 \ \forall i; \qquad |M_1| \geq 2 - .2B_1; \qquad |M_{i+1}| \geq B_i^2 - .2B_{i+1} \ \forall 1 < i < n.$$

As we still have (5), the first inequality above yields $1.2B_i \geq |M_i|$ for all i. Combining this with the second and third inequalities above gives:

$$1.4B_1 \geq 2 \implies B_1 \geq 1.42;$$
$$1.2B_{i+1} \geq |M_{i+1}| \geq B_i^2 - .2B_{i+1} \implies 1.4B_{i+1} \geq B_i^2 \ \forall 1 < i < n.$$

Together, the above yield $B_n \geq 1.4(1.42/1.4)^{2^{n-1}}$, and we again see that exponential bit-complexity is required for a degree-2 SOS proof of $p_n(x, y) \geq -.01$ mod (K). (Incidentally, this also rules out the possibility of "SOS-refuting" the statement $p_n(x, y) < -o_n(1)$ with degree 2.)

2.2 Even with the "Archimedean" constraints

Finally, it's easy to see that the conclusion doesn't change even if we add to (K) the additional constraints $x_i^2 \leq 1$ and $y_i^2 \leq 1$ for all i, making the domain "Archimedean" ("explicitly bounded"). We know these constraints are actually redundant, so still $p_n(x,y)$ has minimal value 0. As for the effect on degree-2 SOS proofs, the new constraints allow us to also add terms $D_i(1-x_i^2)$ and $E_i(1-y_i^2)$ on the RHS of (6) for nonnegative constants D_i, E_i. In turn, this changes (8) to

$$\epsilon = \sum_i (C_i^2 + D_i + E_i), \quad A_i^2 + 2U_i - D_i = 1 \ \forall i, \quad M_1 + 2V_1 = -2,$$
$$M_{i+1} + 2V_{i+1} = -B_i^2 + E_i \ \forall 1 < i < n.$$

The first constraint implies $D_i, E_i \leq .01$ for all i. Given $D_i \leq .01$, we can still deduce $A_i^2 - .2A_i \leq 1.01$, which still implies $A_i \leq 1.2$. The condition $E_i \leq .01$ changes $|M_{i+1}| \geq B_i^2 - .2B_{i+1}$ to $|M_{i+1}| \geq B_i^2 - .2B_{i+1} - .01$, and hence we only get $1.4B_{i+1} \geq B_i^2 - .01$ for all $1 < i < n$. But this is still enough to conclude B_n is doubly-exponential in n, as before. In summary:

▶ **Theorem 1.** *Subject to* (K) *and* $x_i^2 \leq 1$, $y_i^2 \leq 1$, *there is a degree-2 SOS proof that* $p_n(x,y) \geq 0$. *However any degree-2 SOS proof even of* $p_n(x,y) \geq -.01$ *requires bit-complexity* $\Theta(2^n)$.

As a further remark, in the "explicitly bounded" case it's known that there is no SDP duality gap. So instead of trying to use semidefinite programming to get an SOS proof of $p_n(x,y) \geq -o_n(1)$, we might try using it to find a "pseudoexpectation" $\widetilde{\mathbf{E}}[\cdot]$ that satisfies the constraints and minimizes $\widetilde{\mathbf{E}}[p_n(x,y)]$. (See [3] for more on this terminology.) In this dual case, there won't be any "R problem", but instead we'll get an "r problem". The Ellipsoid Algorithm might be used to produce an $\widetilde{\mathbf{E}}[\cdot]$ that satisfies all the constraints up to doubly-exponentially small tolerance; e.g., the *genuine* distribution $x_i \equiv 0$, $y_i \equiv 2^{-2^i}$ satisfies all constraints except for $y_n^2 = 0$, which it satisfies to doubly-exponentially small tolerance. As constructors of SDP hierarchy integrality gaps know, the step of massaging an almost-satisfying solution to an exactly-satisfying solution is often non-obvious and problem-specific.

3 Discussion

I think that Theorem 1 gives a particularly simple example of things going wrong. It's not too much different from, say, the SOS formulation of Maximum Independent Set. Undoubtedly there are generic extensions of the SOS method that will handle this one specific example. For example, the Gröbner basis technique will not have exponential complexity in this case, and will in fact lead to an efficient degree-2 SOS-proof of $p_n(x,y) \geq 0$. We also did not analyze how degree-*four* SOS behaves on this instance. But the point is that it's not so easy to think of generic SOS extensions that will always work in polynomial time. Nor is it easy to think of additional structural constraints on instances that may help, yet that are not too restrictive.

An obvious candidate for additional structure is the constraint that every variable is not just bounded in $[-1,1]$ but *Boolean*. This is at least a common scenario in combinatorial optimization. One still has to be careful though. For example, there is a well-known trick for converting inequality constraints to equality constraints in SOS: replace $q_i(x) \geq 0$ with

$q_i(x) = z^2$ where z is a new variable. However this new variable wouldn't be constrained to be Boolean.

I don't know whether constraining every variable to be Boolean will cause feasible SOS SDPs to always have solutions of polynomial bit-complexity. However in the next section I'll observe that if these are the *only* constraints, we are in good shape. Nevertheless, this seems to me a somewhat rare situation; e.g., it's not satisfied in the Balanced Separator or Independent Set examples. And as noted earlier, although you may succeed in SOS-proving $p(x) \geq \theta$ subject to $x_i^2 = 1\ \forall i$, it's always fundamentally better to try SOS-proving $-1 \geq 0$ subject to the extra constraint $p(x) \leq \theta - \epsilon$. But then you have an additional constraint-inequality in addition to Booleanness.

4 Automatizing SOS proofs subject only to Booleanness

Here I'll record the known observation that, in $n^{O(d)}$ time, we can approximately test if some $p(x)$ is a degree-d sum of squares, modulo $\{x_i^2 = 1,\ \forall i\}$. (To avoid an extra parameter, I'll assume the coefficients of $p(x)$ are rationals expressible with $n^{O(d)}$ bits.) Specifically, if indeed $p(x)$ is degree-d SOS, then the algorithm will find a degree-d SOS representation of $p(x) + \epsilon$ for some $0 \leq \epsilon \leq 2^{-N^{O(d)}}$. I emphasize that there is no mathematical innovation in this section; all the details herein are known.

The typical way to formulate this problem as an SDP is to consider real symmetric matrices X with rows and columns indexed by the $N = n^{O(d)}$ subsets $S \subseteq [n]$, $|S| \leq d/2$. Then $p(x)$ is degree-d SOS mod $\{x_i^2 = 1,\ \forall i\}$ if and only if

$$\exists X \succeq 0 \quad \text{such that} \quad \sum_{\substack{|S|,|T| \leq d/2 \\ S \Delta T = U}} X_{S,T} = p_U \ \forall\ U \subseteq [n],\ |U| \leq d, \tag{SDP}$$

where p_U denotes the coefficient of $p(x)$ on $\prod_{i \in U} x_i$. (We may assume without loss of generality that $p(x)$ is multilinear.) This SDP feasibility problem can be written down using $\mathrm{poly}(N)$ bits, and we want to argue it can be decided (approximately) in $\mathrm{poly}(N)$ time.

The key observation is that the $U = \emptyset$ constraint of (SDP) is precisely "$\mathrm{tr}(X) = p_\emptyset$". The bit-complexity of $p_\emptyset$ is $\mathrm{poly}(N)$, by assumption (in general, it's bounded by the input size). Thus any feasible PSD solution X has the sum of its eigenvalues at most $\mathrm{poly}(N)$, and hence squared Frobenius norm at most $\mathrm{poly}(N)$. This means we can take the "R" in the Ellipsoid Algorithm to be $\mathrm{poly}(N)$, overcoming the main difficulty described in this note. (Note that we could still run in $\mathrm{poly}(N)$ time even if R were $2^{\mathrm{poly}(N)}$.)

We now recall how to take care of the "r" for the Ellipsoid Algorithm. The linear constraints in (SDP) obviously preclude any feasible region from containing a ball of positive radius. So we relax the "$= p_U$" equality constraints to two-sided "$\in [p_U - \epsilon', p_U + \epsilon']$" inequalities, where $\epsilon' = 2^{-N^c}$ for some constant c. (Note that this preserves feasibility, and the bit-complexity of ϵ' is just $\mathrm{poly}(N)$.) Actually, we're still not done because of the additional symmetry requirement $X_{S,T} = X_{T,S}$, but we can take care of this as in the original paper by Grötschel, Lovász, and Schrijver [10] by not introducing variables for the below-diagonal elements of X, treating them implicitly. We can now take the Ellipsoid Algorithm's "r" parameter to be $2^{-\mathrm{poly}(N)}$, as needed.

Finally, we have a $\mathrm{poly}(N)$-time "strong separation oracle" for the relaxed form of (SDP). This follows immediately from the fact that testing whether a matrix of rationals is PSD can be done exactly in polynomial time, as noted in [10].[5] Thus the Ellipsoid Method, as

[5] I've found that the correct proof of this fact appears extremely rarely in the literature; indeed, I've only

described thoroughly in [11], will find a solution to the relaxed form of (SDP) in poly(N) time, provided one exists.

By performing $LDL^\top$ decomposition on the solution (see, e.g., [21]), in poly(N) time we get an exact SOS representation $p'(x) = \sum_{j=1}^{n} c_j r_j(x)^2$, where $p'(x)$ is a polynomial with the property that $|p_U - p'_U| \leq 2^{-N^c}$ for all U. (Here c_j and the coefficients of $r_j(x)$ are rational, and the degree of each $r_j(x)$ is at most $d/2$.) Writing $\Delta(x) = p'(x) - p(x)$, we have a degree-d SOS representation of $p(x) + \Delta(x)$, where Δ has degree at most d and all coefficients bounded by ϵ. Now for each monomial δx^U in $\Delta(x)$, we can get a degree-d SOS proof of $\delta x^U \leq |\delta|$ by using either $x^U = -1 + \frac{1}{2}(x^V + x^W)^2$ or $x^U = 1 - \frac{1}{2}(x^V - x^W)^2$, where V and W partition U into two sets, each of cardinality at most $d/2$. Adding these in for each of the roughly N^2 potential monomials of $\Delta(x)$ therefore gives a degree-$2d$ SOS representation of $p(x) + \epsilon$ for $\epsilon \lessapprox N^2 2^{-N^c}$, and we can make the constant c as large as we want.

5 Conclusion

Several papers have shown that certain "hard-seeming instances" of combinatorial optimization problems — like Unique-Games or Balanced-Separator — are not hard for the constant-degree SOS proof system. Optimistically, this might be evidence that there are better polynomial-time approximation algorithms for the problems than those currently known. But in the end, if we want to show that certain approximation tasks are literally in P in the Turing machine model, we'll have to treat some of the details discussed in this paper.

A good open problem is to establish useful conditions under which this treatment can be done automatically. E.g., does the **Question** from Section 1 have a positive answer if the constraints $x_i^2 \leq 1$ are upgraded to $x_i^2 = 1$?

Regarding errata for my own works: In [22, 14] we showed that certain explicit families of combinatorial optimization instances have low-degree SOS analyses. I did not yet verify that these SOS proofs also have the necessarily small bit-complexity that would allow an efficient algorithm to (approximately) find them. However we didn't formally claim that these algorithms exist; the theorems in these works were just evidence that SOS *might* be successful on all instances. In [2] we claimed that SOS algorithms could efficiently refute random instances of certain CSPs with certain parameters. I'm confident that this statement is true, but rather than prove it I'll simply say that the SOS-ability here is essentially just a side comment. It's clear that there is *some* efficient algorithm: all that's ultimately needed for refutation is the certification that a certain symmetric matrix A has $\|A\| \leq O(\text{small})$. This is equivalent to $O(\text{small}) \cdot I - A \succeq 0$, and as noted in Section 4, testing semidefiniteness of rational matrices can be done efficiently.

Acknowledgments. I would like to thank Boaz Barak, Sangxia Huang, Etienne de Klerk, Pravesh Kothari, James Lee, Pablo Parrilo, Dima Pasechnik, David Steurer, and especially

seen it in the work of Grötschel, Lovász, and Schrijver [10, 11] and in a survey article by Lovász [19]. You certainly can't just "compute the eigenvalues of the matrix and check if they're nonnegative". It's also a somewhat common misconception [29, 4] that X is PSD if and only if its N leading principal minors are nonnegative. The correct proof from [19] involves seeing whether the Cholesky (LDL) decomposition on X succeeds. (This is essentially the same as the proof in [10], which involves finding the image of X, then checking if X is strictly positive definite on the image by testing if the leading principal minors are strictly positive.) In turn, this relies on the old but nonobvious [6] fact due to Edmonds [5] that Gaussian Elimination is in polynomial time.

David Witmer for discussions. I would also like to thank Dieter van Melkebeek for pointing out the incorrect justification of (4) in an earlier draft of this work.

References

1. Farid Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5(1):13–51, 1995.
2. Sarah Allen, Ryan O'Donnell, and David Witmer. How to refute a random CSP. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, 2015.
3. Boaz Barak and David Steurer. Sum-of-squares proofs and the quest toward optimal algorithms. In *Proceedings of the 2014 International Congress of Mathematicians*. International Mathematical Union, 2014.
4. Charles Delorme and Svatopluk Poljak. Laplacian eigenvalues and the maximum cut problem. *Mathematical Programming*, 62(1–3):557–574, 1993.
5. Jack Edmonds. Systems of distinct representatives and linear algebra. *Journal of Research of the National Bureau of Standards*, 71B:241–245, 1967.
6. Jeff Erickson. What is the actual time complexity of Gaussian elimination? `http://cstheory.stackexchange.com/questions/3921/what-is-the-actual-time-complexity-of-gaussian-elimination`, 2010.
7. Dima Grigoriev. Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity. Technical Report IHES/M/99/68, Insitut des Hautes Études Scientifiques, 1999.
8. Dima Grigoriev. Complexity of Positivstellensatz proofs for the knapsack. *Computational Complexity*, 10(2):139–154, 2001.
9. Dima Grigoriev and Nicolai Vorobjov. Complexity of Null- and Positivstellensatz proofs. *Annals of Pure and Applied Logic*, 113(1):153–160, 2001.
10. Martin Grötschel, László Lovász, and Alexander Schrijver. The Ellipsoid Method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.
11. Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer–Verlag, 1988.
12. Amir Hashemi and Daniel Lazard. Sharper complexity bounds for zero-dimensional gröbner bases and polynomial system solving. *International Journal of Algebra and Computation*, 21(05):703–713, 2011.
13. Cédric Josz and Didier Henrion. Strong duality in Lasserre's hierarchy for polynomial optimization. *Optimization Letters*, 10(1):3–10, 2016.
14. Manuel Kauers, Ryan O'Donnell, Li-Yang Tan, and Yuan Zhou. Hypercontractive inequalities via SOS, and the Frankl-Rödl graph. *Discrete Analysis*, 4, 2016.
15. Leonid Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53–72, 1980.
16. Jean Lasserre. Optimisation globale et théorie des moments. *Comptes Rendus de l'Académie des Sciences*, 331(11):929–934, 2000.
17. Jean Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.
18. Monique Laurent. Sums of squares, moment matrices and optimization over polynomials. *Emerging Applications of Algebraic Geometry*, 149:157–270, 2009.
19. László Lovász. Semidefinite programs and combinatorial optimization. In *Recent advances in algorithms and combinatorics*, pages 137–194. Springer New York, 2003. `doi:10.1007/0-387-22444-0_6`.
20. Yurii Nesterov. *Squared functional systems and optimization problems*, chapter 17, pages 405–440. Kluwer Academic Publishers, 2000.

21 Ryan O'Donnell and Franklin Ta. Linear programming and semidefinite programming lecture 10 notes, 2011. http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15859-f11/www/notes/lecture10.pdf.

22 Ryan O'Donnell and Yuan Zhou. Approximability and proof complexity. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1537–1556, 2013.

23 Pablo Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, 2000.

24 Lorant Porkolab and Leonid Khachiyan. On the complexity of semidefinite programs. *Journal of Global Optimization*, 10(4):351–365, 1997.

25 Motakuri Ramana. An exact duality theory for semidefinite programming and its complexity implications. *Mathematical Programming*, 77(1):129–162, 1997.

26 Claus Scheiderer. Sums of squares of polynomials with rational coefficients. *Journal of the European Mathematical Society*, 18(7):1495–1513, 2016.

27 Grant Schoenebeck. Linear level Lasserre lower bounds for certain k-CSPs. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 593–602, 2008.

28 Naum Shor. Class of global minimum bounds of polynomial functions. *Cybernetics*, 23(6):731–734, 1987.

29 Kuduvally Swamy. On Sylvester's criterion for positive-semidefinite matrices. *IEEE Transactions on Automatic Control*, 18(3):306–306, 1973.

30 Sergey Tarasov and Mikhail Vyalyi. Semidefinite programming and arithmetic circuit evaluation. *Discrete Applied Mathematics*, 156(11):2070–2078, 2008.

The Journey from NP to TFNP Hardness*

Pavel Hubáček[1], Moni Naor[2], and Eylon Yogev[3]

1 **Weizmann Institute of Science, Rehovot, Israel**
pavel.hubacek@weizmann.ac.il
2 **Weizmann Institute of Science, Rehovot, Israel**
moni.naor@weizmann.ac.il
3 **Weizmann Institute of Science, Rehovot, Israel**
eylon.yogev@weizmann.ac.il

Abstract

The class TFNP is the search analog of NP with the additional guarantee that any instance has a solution. TFNP has attracted extensive attention due to its natural syntactic subclasses that capture the computational complexity of important search problems from algorithmic game theory, combinatorial optimization and computational topology. Thus, one of the main research objectives in the context of TFNP is to search for efficient algorithms for its subclasses, and at the same time proving hardness results where efficient algorithms cannot exist.

Currently, no problem in TFNP is known to be hard under assumptions such as NP hardness, the existence of one-way functions, or even public-key cryptography. The only known hardness results are based on less general assumptions such as the existence of collision-resistant hash functions, one-way permutations less established cryptographic primitives (e.g., program obfuscation or functional encryption).

Several works explained this status by showing various barriers to proving hardness of TFNP. In particular, it has been shown that hardness of TFNP hardness cannot be based on worst-case NP hardness, unless NP = coNP. Therefore, we ask the following question: What is the weakest assumption sufficient for showing hardness in TFNP?

In this work, we answer this question and show that hard-on-average TFNP problems can be based on the weak assumption that there exists a *hard-on-average* language in NP. In particular, this includes the assumption of the existence of one-way functions. In terms of techniques, we show an interesting interplay between problems in TFNP, derandomization techniques, and zero-knowledge proofs.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases TFNP, average-case hardness, one-way functions, derandomization, zero-knowledge

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.60

1 Introduction

The class NP captures all *decision* problems for which the "yes" instances have efficiently verifiable proofs. The study of this class and its computational complexity are at the heart of theoretical computer science. Part of the effort has been to study the *search* analog of NP which is defined by the class FNP (for Function NP) and captures all search problems for which verifying a solution can be done efficiently. Megiddo and Papadimitriou [43] introduced

* Supported in part by a grant from the I-CORE Program of the Planning and Budgeting Committee, the Israel Science Foundation and BSF.

licensed under Creative Commons License CC-BY
8th Innovations in Theoretical Computer Science Conference (ITCS 2017).
Editor: Christos H. Papadimitrou; Article No. 60; pp. 60:1–60:21

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the complexity class TFNP (for Total Function NP) which is a subclass of FNP with the additional property of the problem being *total*, i.e., for any instance a solution is guaranteed to exist. For this reason, the class TFNP is usually considered as the search variant containing NP ∩ coNP: in particular, for any language $L \in \mathsf{NP} \cap \mathsf{coNP}$, the corresponding search problem is given by an instance x and a solution is either a witness verifying that $x \in L$ or a witness verifying that $x \notin L$. Since $L \in \mathsf{NP} \cap \mathsf{coNP}$, such a solution must always exist.

Beyond its natural theoretical appeal, TFNP attracted extensive attention due to its important syntactic subclasses. The common way of defining such subclasses is via various non-constructive arguments used for proving totality of search problems. For example, the *parity argument for directed graphs*: "If a directed graph has an unbalanced node (a vertex with unequal in-degree and out-degree), then it must have another unbalanced node," gives rise to the class PPAD (for Polynomial Parity Argument on Directed graphs [47]). This might be the most famous subclass of TFNP due to the fact that one of its complete problems is finding Nash equilibria in strategic games [19, 18]. Other known subclasses are PPP (for Polynomial Pigeonhole Principle [47]), PLS (for Polynomial Local Search [36]), and CLS (for Continuous Local Search [20]).

It is easy to see that if P = NP then all search problems in TFNP can be solved efficiently. Therefore, the study of the hardness of TFNP classes must rely on some hardness assumptions (until the $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$ question is resolved). A related issue is to establish hard distributions for problems in TFNP. Here it is natural to use hardness of cryptographic assumptions, and therefore the goal is to base TFNP hardness on different cryptographic primitives. For instance, the (average-case) hardness of the subclass PPP has been based on the existence of either one-way permutations[1] [47] or collision-resistant hash[2] [35]. For other subclasses even less standard assumptions have been used. The hardness of PPAD, PLS and even CLS (a subclass of their intersection) has been based on strong cryptographic assumptions, e.g., indistinguishability obfuscation and functional encryption [13, 26, 29].

To understand the hierarchy of cryptographic assumptions it is best to turn to Impagliazzo's worlds [30]. He described five possible worlds: `Algorithmica` (where P = NP), `Heuristica` (where NP is hard in the worst case but easy on average, i.e., one simply does not encounter hard problems in NP), `Pessiland` (where hard-on-average problems in NP exist, but one-way functions do not exist), `Minicrypt` (where one-way functions exist[3]), and `Cryptomania` (where Oblivious Transfer exists[4]). Nowadays, it is possible to add a sixth world, `Obfustopia` where indistinguishability obfuscation for all of P is possible [7, 25, 51]. Our goal is to connect these worlds to the world of TFNP.

So far the hardness of TFNP was only based either on `Obfustopia` or strong versions of `Minicrypt` (e.g. one-way permutations). None of the assumptions used to show TFNP hardness are known to be implied simply by the existence of one-way functions, or even from public-key encryption. It is known that one-way permutations cannot be constructed in a black-box way from one-way functions [50, 37]. Moreover, indistinguishability obfuscation is a relatively new notion that has yet to find solid ground (see [1, Appendix A] for a summary

[1] A permutation is one-way if it is easy to compute on every input, but hard to invert on a random image. Such permutations exist only if $\mathsf{P} \neq \mathsf{NP} \cap \mathsf{coNP}$ [16].

[2] A collision-resistant hash is a hash function such that it is hard to find two inputs that hash to the same output. Simon [52] showed a *black-box separation* between one-way functions and collision-resistant hashing.

[3] For many primitives such as shared-key encryption, signatures, and zero-knowledge proofs for NP it is known that their existence is equivalent to the existence of one-way functions.

[4] Roughly speaking, this world is where public-key cryptography resides.

of known attacks as of August 2016). Moreover, while indistinguishability obfuscation implies the existence of one-way functions [39], it has been shown that it cannot be used to construct either one-way permutations or collision-resistant hash in a black-box manner [5, 6]. Other hardness results for TFNP rely on specific number theoretic assumptions such as factoring or discrete log.

Barriers for proving TFNP hardness. Given the lack of success in establishing hardness of TFNP under general assumptions, there has been some effort in trying to explain this phenomenon. From the early papers on PLS and TFNP by Johnson et al. [36] and Megiddo and Papadimitiriou [43] we know that TFNP hardness cannot be based on worst-case NP hardness, unless NP = coNP.[5] Mahmoody and Xiao [42] showed that even a *randomized* reduction from TFNP to worse-case NP would imply that SAT is "checkable" (which is a major open question [15]). Buhrman et al. [17] exhibited an oracle under which all TFNP problems are easy but the polynomial hierarchy is infinite, thus, explaining the failure to prove TFNP hardness based on worst-case problems in the polynomial hierarchy. In a recent work, Rosen et al. [49] showed that any attempt to base TFNP hardness on (trapdoor) one-way functions (in a black-box manner) must result in a problem with exponentially many solutions. This leads us to ask the following natural question:

What is the weakest assumption under which we can show hardness of TFNP*?*

A possible approach to answering this question is to try to put TFNP hardness in the context of the Impagliazzo's five worlds. In the light of this classification, one can argue that an equally, if not more interesting, question is:

What is the weakest assumption for hardness on average of TFNP*?*

In this work, we give an (almost) tight answer to this question. Given the negative results discussed above, our results are quite unexpected. While the barriers imply that it is very unlikely to show TFNP hardness under *worst-case* NP hardness, we are able to show that *hard-on-average* TFNP can be based on any *hard-on-average* language in NP (and in particular on the existence of one-way functions). In the terminology of the worlds of Impagliazzo, our results show that hard-on-average TFNP problems exist in `Pessiland` (and beyond), a world in which none of the assumptions previously used to show TFNP hardness exist (not even one-way functions). On the other hand, the barriers discussed above indicate that it would be unlikely to prove worse-case TFNP hardness in `Heuristica`, as in that world the only available assumption is $\mathsf{P} \neq \mathsf{NP}$.

As for techniques, we show an interesting interplay between TFNP and derandomization. We show how to "push" problems into TFNP while maintaining their hardness using derandomization techniques. In the non-uniform settings, our results can be established with no further assumptions. Alternatively, we show how to get (standard) uniform hardness via further derandomization assuming the existence of a Nisan-Wigderson type pseudorandom generator (discussed below).

In correspondence with the black-box impossibility of Rosen et al. [49], the problem we construct might have instances with an exponential number of solutions. Nevertheless, we

[5] The argument is quite simple: Suppose there is a reduction from SAT (for example) to TFNP, in a way that every solution to the TFNP problem translates to a resolution of the SAT instance. This would yield an efficient way to refute SAT: guess a solution of the TFNP problem, and verify that it indeed implies that the SAT instance has no solution.

show that there exists a different problem in TFNP such that its hardness can be based on the existence of one-way functions and *Zaps* (discussed below), and has *either one or two* solutions for any instance. The reason our result does not contradict the impossibility results is (i) that it is not black-box, and (ii) we use an extra object, Zaps; it is unknown whether the latter exists solely under the assumption that one-way functions exist (but it can be shown to exist relative to random oracles). We observe that the fact that our problem has either one or two solutions for every instance is (in some sense) tight: any hard problem with a single solution for every instance would imply hardness of $\mathsf{NP} \cap \mathsf{coNP}$ (by taking a hardcore bit of the unique solution), and thus would have to face the limitations of showing $\mathsf{NP} \cap \mathsf{coNP}$ hardness from unstructured assumptions [12].

Nisan-Wigderson PRGs. Nisan and Wigderson showed how the assumption of a very hard language can be used to construct a specific type of pseudorandom generators (henceforth NW-type PRG) benefitial for tasks of derandomization and in particular to derandomize BPP algorithms [46]. Impagliazzo and Wigderson [34] constructed a NW-type PRG under the (relatively modest) assumption that E (i.e., $\mathsf{DTIME}(2^{O(n)})$) has a problem of circuit complexity $2^{\Omega(n)}$. Although used mainly in computational complexity, (strong versions of) these generators have found applications in cryptography as well: Barak, Lindell, and Vadhan [8] used them to prove an impossibility for two-round zero-knowledge proof systems, Barak, Ong, and Vadhan [9] showed how to use them to construct a witness indistinguishable proof system for NP, and Bitansky and Vaikuntanathan [14] showed how to completely immunize a large class of cryptographic algorithms from making errors. In the examples above, the PRG was constructed to fool polynomial sized (co)non-deterministic circuits. Such NW-type PRGs follow from (relatively modest) assumption that E has a problem of (co)non-deterministic circuit complexity $2^{\Omega(n)}$.

We show that NW-type PRGs have an interesting interplay with TFNP as well. We use strong versions of these generators to show that several different problems can be "pushed" into TFNP by eliminating instances with no solutions (in the uniform setting). Our notion of strong NW-type PRG requires fooling polynomial-sized Π_2-circuits. Again, such PRGs follow from the assumption that E has a problem of Π_2-circuit complexity $2^{\Omega(n)}$ (see [4] for an example of a use of such PRGs).

Zaps. Feige and Shamir [24] suggested a relaxed notion of zero-knowledge called *witness indistinguishability* where the requirement is that any two proofs generated using two different witnesses are computationally indistinguishable. They showed how to construct three-message witness indistinguishable proofs for any language in NP assuming the existence of one-way functions. A Zap, as defined by Dwork and Naor [21], is a two-message public-coin witness indistinguishable scheme where the first message can be reused for all instances. They showed that (assuming one-way functions) Zaps are existentially equivalent to NIZKs (non-interactive zero-knowledge proofs), and hence one can use the known constructions (e.g., based on trapdoor permutations). Dwork and Naor also showed that the interaction could be further reduced to a single message witness indistinguishable proof system in the non-uniform setting (i.e., the protocol has some polynomial sized advice). In the uniform setting, Barak et al. [9] showed the same result by leveraging a NW-type PRG for derandomizing the known constructions of Zaps. Our proofs make use of such witness indistinguishable proof systems both in the uniform and non-uniform setting.

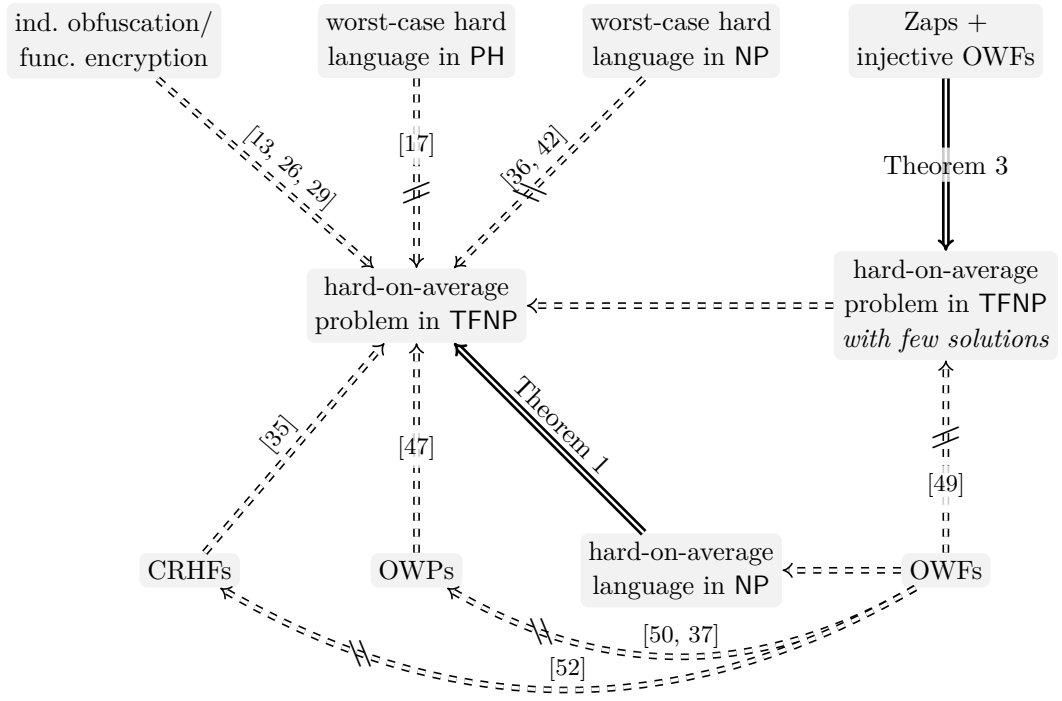


Figure 1 An illustration of our results (solid implications) in the context of previously known positive results (dashed implications) and negative results (crossed out dashed implications).

1.1 Our results

Some of our results use a derandomization assumption in the form "assume that there exists a function with deterministic (uniform) time complexity $2^{O(n)}$, and Π_2-circuit complexity $2^{\Omega(n)}$". In the description of our results below, we simply call this the *fooling assumption*. Alternatively, instead of this assumption we can consider the non-uniform setting, and get the same results for TFNP/poly (see Definition 10). Some of our results are summarized in Figure 1.

▶ **Theorem 1** (Informal). *Any hard-on-average* NP *language (e.g., random SAT, hidden clique, or any one-way function) implies a non-uniform* TFNP *problem which is hard-on-average.*

Then, using derandomization we get the following corollary for the uniform setting.

▶ **Corollary 2** (Informal). *Under the fooling assumption, any hard-on-average* NP *language implies a (uniform)* TFNP *problem which is hard-on-average.*

Furthermore, we present an alternative approach for proving totality of search problems using zero-knowledge proofs which allows to build more structured TFNP problems. Specifically, if injective one-way functions exist, and Zaps exist then we construct a total search problem with at most two solutions.

▶ **Theorem 3** (Informal). *Assume the existence of Zaps and injective one-way functions. Then, there exists a hard-on-average problem (either non-uniform, or uniform under the fooling assumption) in* TFNP *such that any instance has at most two solutions.*

1.2 Related work (search vs. decision)

The question of the complexity of search when the solution is guaranteed to exist has been discussed in various scenarios. Even et al. [23] considered the complexity of promise problems and the connection to cryptography. They raised a conjecture which implies that public-key cryptography based on NP-hard problems does not exist. Impagliazzo and Naor [33] and Lovasz et al. [41] considered search in the query complexity model where a solution is guaranteed to exist, similarly to the class TFNP. They showed a separation between the classes of deterministic, randomized, and the size of the object of the search. Bellare and Goldwasser [10] considered the issue of self reducibility, i.e. are there languages in NP where given a decision oracle it is hard to solve *any* corresponding FNP search problem. They showed that such languages exist under the assumption that EE $\neq$ NEE (double exponential time is not equal the non-deterministic version of it).

2 Our Techniques

2.1 TFNP hardness based on average-case NP hardness

Let L be a hard-on-average NP language for an efficiently samplable distribution $\mathcal{D}$, associated with a relation R_L. That is, $(L, \mathcal{D})$ is a pair such that no efficient algorithm can decide L with high probability when instances are drawn from $\mathcal{D}$. Our goal is to construct a (randomized) reduction from an instance for L sampled from $\mathcal{D}$ to a TFNP problem, such that every solution to the TFNP problem can be used to decide the instance given for L. As discussed above, such a reduction cannot rely on worst-case complexity ([36, 43]), and hence it must use the fact that we have a hard distribution for L.

The first thing to note is that since $(L, \mathcal{D})$ is hard to decide, then it is also hard to find a valid witness. Thus, the corresponding search problem, i.e., given an instance x to find a witness w such that $(x, w) \in R_L$, is hard directly based on the hardness of $(L, \mathcal{D})$. However, this is not sufficient for establishing hardness in TFNP. The issue is that not all instances indeed have a solution, and furthermore, determining whether an instance has a solution is by itself NP-hard.

To show that a problem is in TFNP we need to prove that there exists a solution for *every* instance. Therefore, our goal is to start with $(L, \mathcal{D})$ and end up with $(L', \mathcal{D}')$, such that the distribution $\mathcal{D}'$ always outputs instances in L', but remains a hard distribution nevertheless. We employ a method called *reverse randomization*, which has been used in the context of complexity (for proving that BPP is in the polynomial hierarchy [40]) and in the context of cryptography (for constructing Zaps, commitment schemes, and immunizing cryptographic schemes from errors [44, 21, 22, 14]). For a string s, let the distribution $\mathcal{D}_s$ be the one that on random coins r samples both $x \leftarrow \mathcal{D}(1^n; r)$ and $x' \leftarrow \mathcal{D}(1^n; r \oplus s)$. We define L_s accordingly as containing the instance (x, x') if *at least one* of x or x' is in L. Since D is a hard distribution, we know that for x' sampled from D the probability that $x' \in L$ is non-negligible. Therefore, we get that for any instance $x \notin L$ with non-negligible probability over the choice of s the shifted version x' of x is such that $x' \in L$. We perform several such random shifts, such that for any $x \notin L$ the probability that *one* of its shifts is an element in L is very high, and define the new language to accept any instance where *at least one of the shifts* is in L. Moreover, we show that for any such collection of shifts, the resulting distribution is still hard even when the shift is given to the solving algorithm.

The result is that there *exists* a collection of shifts such that applying them to $\mathcal{D}$ yields a pair $(L', \mathcal{D}')$ with the following two properties: (1) the support of $\mathcal{D}'$ is entirely contained in

L', and (2) the pair $(L', \mathcal{D}')$ is a hard *search* problem, i.e., given a random instance x sampled from $\mathcal{D}'$ the probability of finding a valid witness for x in polynomial-time is negligible. To construct our hard TFNP problem there are a few difficulties we ought to address.

We have merely proven the existence of such a shift for every input length. However, finding such a shift might be a hard task on its own. One possible way to circumvent this issue is to consider non-uniform versions of the problem, where this shift is hardwired into the problem's description (for each input size). Notice that in this case the shift is public and thus it is important that we have proven hardness even given the shift. If we want to stay in the uniform setting, we need to show how to find such shifts efficiently for every n. The main observation here is that we can show that a random shift will be good enough with high probability, and verifying if a given shift is good can be done in the complexity class Π_2. Thus, using a Nisan-Wigderson type pseudorandom generators against Π_2-circuits we show how to (completely) derandomize this process and find the collection of shifts efficiently and deterministically. Such NW-type PRGs were shown to exist under the assumption that E is hard for Π_2 circuits.

At this point, we have an (efficiently) samplable distribution $\mathcal{D}'$ such that its support provably contains only instances in L', and finding any valid witness is still hard for any polynomial-time algorithm. However, how can we use $\mathcal{D}'$ to create a hard TFNP problem? Indeed, we can use $(L', \mathcal{D}')$ to construct a hard search problem. Since any instance in the support of $\mathcal{D}'$ is satisfiable we would claim that the problem is indeed in TFNP. However, this is actually not the case, since it is infeasible to verify that an instance has actually been sampled from $\mathcal{D}'$! Therefore, the resulting search problem is not in TFNP. To solve this, we need a way to prove that an instance x is in the support of $\mathcal{D}'$ without hurting the hardness of the distribution.

On distributions with public randomness. The straightforward solution to the above problem is to publish the randomness r used to sample the instance. This way, it is easy to verify that the instance is indeed in the support of $\mathcal{D}'$ and thus any instance must have a solution and our problem is in TFNP. But what about hardness? Is the distribution hard even when given the randomness used to sample from it? Note that in many cases the randomness might even explicitly contain a satisfying assignment (e.g., a planted clique or factorization of a number). Thus, we must rely on distributions which remain hard even when given the random coins used for sampling. We denote such distributions as *public-coin distributions*.

If the original distribution $\mathcal{D}$ was public-coin to begin with, then we show that our modifications maintain this property. Thus, assuming that $\mathcal{D}$ is public-coin we get a problem that is provably in TFNP and is as hard as $\mathcal{D}$ (see Section 4.1 for the full proof). Then, we show that in fact any hard distribution can be modified to construct a public-coin distribution while maintaining its hardness, with no additional assumptions. This lets us construct a hard TFNP problem using any hard-on-average language L, even one with a distribution that is not public-coin (see Section 4.2 for discussion of the transformation).

The number of solutions of the TFNP problem we constructed is polynomial in the number of witnesses of the language L we begin with (each shift introduces new solutions). In particular, if we start from a hard decision problem $(L, \mathcal{D})$ where $\mathcal{D}$ is public-coin and every $x \in L$ has a single witness, then our TFNP problem will have only a polynomial number of witnesses. Our transformation from any distribution to a public-coin one increases the number of witnesses (for some instances). In the next section, we discuss an alternative method for proving totality of search problems which, moreover, results in a small number of solutions.

2.2 Using zero-knowledge for TFNP hardness

We demonstrate the power of zero-knowledge proofs in the context of TFNP in order to assure totality of search problems. This enables us to get more structured problems in TFNP.

Suppose we have a one-way function f, and we try to construct a hard TFNP problem where any solution can be used to invert f on a random challenge y. The difficulty is that we need to prove that the search problem is total while not ruining its hardness. In particular, we have to prove that given y there exists an inverse x such that $f(x) = y$ without "revealing anything" beyond the existence of x. If f is a permutation this task is trivial, since every y has an inverse by definition. However, for a general one-way function this is not the case.

To solve this issue, we employ as our main tool a Zap: a two-message witness indistinguishable proof system for NP (discussed in Section 1). We construct a problem where an instance contains an image y and a Zap proof π of the statement "y has an inverse under f". The first message of the proof is either non-uniformly hardwired in the problem description, or uniformly derandomized (see discussion in the introduction). This way, any instance has a solution: if π is a valid proof then by the perfect soundness of the Zap we know that y indeed has an inverse, and otherwise, we consider the all-zero string to be a solution. Our goal is to show that this problem is still hard. However, the Zap proof only guarantees that for any two x, x' the proof π is indistinguishable, which does not suffice. In fact, if f is injective, then y has only a single inverse and π might simply be the inverse x without violating the witness indistinguishability property of the Zap.

Therefore, an instance of our problem will consist of two images y, y' and a Zap proof π that *one* of the images has an inverse under f. The goal is to find an inverse of y or y', where one is guaranteed to exist as long as the proof π is valid. This way, we are able to ensure that the proof π does not reveal any useful information about the inverse x. Finally, by randomly embedding an instance y of a challenge to f we show that any adversary that solves this problem can be used to invert the one-way function f. Thus, we get a total problem that is hard assuming one-way functions, and Zaps. Moreover, notice that when the underlying one-way function is *injective* the problem we constructed has *exactly one or two solutions* for any instance. See Theorem 21 and Section 5 for details of the full proof.

3 Preliminaries

We present the basic definitions and notation used in this work. For a distribution X we denote by $x \leftarrow X$ the process of sampling a value x from the distribution X. For an integer $n \in \mathbb{N}$ we denote by $[n]$ the set $\{1, \ldots, n\}$. A function $\mathsf{neg}\colon \mathbb{N} \to \mathbb{R}$ is negligible if for every constant $c > 0$ there exists an integer N_c such that $\mathsf{neg}(n) < n^{-c}$ for all $n > N_c$.

▶ **Definition 4** (Computational Indistinguishability)**.** Two sequences of random variables $X = \{X_n\}_{n\in\mathbb{N}}$ and $Y = \{Y_n\}_{n\in\mathbb{N}}$ such that X_n's and Y_n's lengths are polynomially bounded are computationally indistinguishable if for every probabilistic polynomial time algorithm A there exists an integer N such that for all $n \geq N$,

$$|\Pr[A(X_n) = 1] - \Pr[A(Y_n) = 1]| \leq \mathsf{neg}(n) \ ,$$

where the probabilities are over X_n, Y_n and the internal randomness of A.

▶ **Definition 5** (One-Way Functions)**.** A polynomial time computable function $f : \{0,1\}^* \to \{0,1\}^*$ is called *one-way* if for any PPT inverter A (the non-uniform version is polysize circuit) there exists a negligible function $\mu(\cdot)$, such that

$$\Pr\left[A(f(x)) \in f^{-1}(f(x)) : x \leftarrow \{0,1\}^n\right] \leq \mu(n) \ .$$

We say that $\mathcal{F}$ is a family of injective one-way functions if every function in $\mathcal{F}$ is injective.

3.1 Average-case complexity

For a language L and an instance x we write $L(x) = 1$ if $x \in L$ and $L(x) = 0$ otherwise.

▶ **Definition 6** (Probability distributions)**.** A probabilistic randomized algorithm $\mathcal{D}$ is said to be a probability distribution if on input 1^n it outputs a string of length n. We denote by $\mathcal{D}(1^n; r)$ the evaluation of $\mathcal{D}$ on input 1^n using the random coins r. We say that $\mathcal{D}$ is efficient if it runs in polynomial time.

▶ **Definition 7** (Hard distributional problem)**.** Let $L \in \mathsf{NP}$ and let $\mathcal{D}$ be an efficient probability distribution. We say that $(L, \mathcal{D})$ is a hard distributional problem if for any probabilistic polynomial time-algorithm A there exist a negligible function $\mathsf{neg}(\cdot)$ such that for all large enough n it holds that:

$$\Pr_{r,A}[A(x) = L(x) : x \leftarrow \mathcal{D}(1^n; r)] \leq 1/2 + \mathsf{neg}(n) \ ,$$

where the probability is taken over r and the randomness of A.

▶ Remark. We say that a language $L \in \mathsf{NP}$ is hard-on-average if there exists an efficient probability distribution $\mathcal{D}$ such that $(L, \mathcal{D})$ is a hard distributional problem.

▶ **Definition 8** (Hard public-coin distributional problem)**.** Let $L \in \mathsf{NP}$ and let $\mathcal{D}$ be an efficient probability distribution. We say that $(L, \mathcal{D})$ is a hard public-coin distributional problem if for any probabilistic polynomial time-algorithm A there exists a negligible function $\mathsf{neg}(\cdot)$ such that for all large enough n it holds that:

$$\Pr_{r,A}[A(r, x) = L(x) : x \leftarrow \mathcal{D}(1^n; r)] \leq 1/2 + \mathsf{neg}(n) \ ,$$

where the probability is taken over r and the randomness of A. (Notice that in this case, A gets both the instance x and the random coins r used to sample x.)

3.2 Total search problems

The class TFNP of "total search problems" contains a host of non-trivial problems for which a solution always exists.

▶ **Definition 9** (TFNP)**.** A *total* NP *search problem* is a relation $\mathcal{S}(x, y)$ such that it is (i) computable in polynomial (in $|x|$ and $|y|$) time (ii) *total*, i.e. there is a polynomial q such that for every x there exists a y such that $\mathcal{S}(x, y)$ and $|y| \leq q(|x|)$.

The set of all total NP search problems is denoted by TFNP.

The class $\mathsf{TFNP/poly}$ is the non-uniform circuit version of TFNP, similar to $\mathsf{NP/poly}$ with respect to NP.

▶ **Definition 10** ($\mathsf{TFNP/poly}$)**.** A *total* $\mathsf{NP/poly}$ *search problem* is a relation $\mathcal{S}(x, y)$ such that it is (i) computable polynomial (in $|x|$ and $|y|$) time with polynomial advice or equivalently there exists a family of polynomial sized circuits that computes $\mathcal{S}$ (ii) *total*, i.e. there is a polynomial q such that for every x there exists a y such that $\mathcal{S}(x, y)$ and $|y| \leq q(|x|)$.

The set of all total $\mathsf{NP/poly}$ search problems is denoted by $\mathsf{TFNP/poly}$.

3.3 Witness indistinguishable proof systems

We consider witness indistinguishable proof systems for NP. In particular, these are derandomized versions of Zaps and can be constructed from any Zap by fixing the first message non-uniformly (see [21, Section 3]). In the uniform setting, Barak et al. [9] showed that by leveraging a NW-type PRG, they can derandomize the Zap construction and get the same result. In both cases, we get a witness indistinguishable proof system which is defined as follows:

▶ **Definition 11.** Let L be an NP language with relation R. A scheme $(\mathsf{Prove}, \mathsf{Verify})$ is a witness indistinguishable proof system between a verifier and a prover if:

1. **Completeness**: for every $(x, w) \in R$ we have that:

$$\Pr\left[\mathsf{Verify}(x, \pi) = 1 \mid \pi \leftarrow \mathsf{Prove}(x, w)\right] = 1 \ .$$

2. **Perfect Soundness**: for every $x \notin L$ and for every π it holds that:

$$\Pr\left[\mathsf{Verify}(x, \pi) = 1\right] = 0 \ .$$

3. **Witness Indistinguishability**: for any sequence of $\{x, w, w'\}$ such that $(x, w) \in R$ and $(x, w') \in R$ it holds that:

$$\{\mathsf{Prove}(x, w)\} \approx_c \{\mathsf{Prove}(x, w')\} \ .$$

3.4 Nisan-Wigderson type pseudorandom generators

We define Nisan-Wigderson type pseudorandom generators [46] that fool circuits of a given size.

▶ **Definition 12** (NW-type PRGs.)**.** A function $\mathsf{G}\colon \{0,1\}^{d(n)} \to \{0,1\}^n$ is an *NW-type* PRG *against circuits of size* $t(n)$ if it is (i) computable in time $2^{O(d(n))}$ and (ii) any circuit C of size at most $t(n)$ distinguishes $U \leftarrow \{0,1\}^n$ from $\mathsf{G}(s)$, where $s \leftarrow \{0,1\}^{d(n)}$, with advantage at most $1/t(n)$.

▶ **Theorem 13** ([34])**.** *Assume there exists a function in* $\mathsf{E} = \mathsf{DTIME}(2^{O(n)})$ *with circuit complexity* $2^{\Omega(n)}$*. Then, for any polynomial* $t(\cdot)$*, there exists a NW-type generator* $\mathsf{G}\colon \{0,1\}^{d(n)} \to \{0,1\}^n$ *against circuits of size* $t(n)$*, where* $d(n) = O(\log n)$*.*

Note that one can find a specific function f satisfying the above condition. In general, any function that is E-complete under linear-time reductions will suffice, and in particular, one can take the bounded halting function.[6]

The above theorem was used in derandomization to fool polynomial sized circuits. It was observed in [4] that Theorem 13 can be extended to more powerful circuits such as non-uniform circuits. In particular, they gave the following theorem that is used in this work.

▶ **Definition 14** (oracle circuits and Σ_i/Π_i-circuits)**.** Given a boolean function $f(\cdot)$, an f-circuit is a circuit that is allowed to use f gates (in addition to the standard gates). A Σ_i-circuit (resp., Π_i-circuit) is an f-circuit where f is the canonical Σ_i^p-complete (resp., Π_i^p-complete) language. The size of all circuits is the total number of wires and gates (see [3, Chapter 5] for a formal definition of the classes Σ_i^p and Π_i^p).

[6] The function is defined as follows: $\mathrm{BH}(M, x, t) = 1$ if the Turing machine M outputs 1 on input x after at most t steps (where t is given in binary), and $\mathrm{BH}(M, x, t) = 0$ otherwise.

▶ **Theorem 15** (cf., [4, Theorem 1.7]). *For every $i \geq 0$, the statement of Theorem 13 also holds if we replace every occurrence of the word "circuits" by "Σ_i-circuits" or alternatively by "Π_i-circuits".*

The assumption underlying the above theorem is a worst-case assumption and it can be seen as a natural generalization of the assumption that $\mathsf{E} \subsetneq \mathsf{NP}$. For a further discussion about this type of assumptions see [4, 2].

4 TFNP Hardness from Average-Case NP Hardness

We show that there exists a search problem in TFNP that is hard-on-average under the assumption of existence of a hard-on-average NP language and a Nisan-Wigderson type complexity assumption. An overview of the proof is given in Section 2.1. Formally, we prove:

▶ **Theorem 16.** *If there exists a hard-on-average language in* NP *then there exists a hard-on-average problem in non-uniform* TFNP.

Under an additional (worst-case) complexity assumption (as discussed in Section 1.1), we also give a uniform version of this result.

▶ **Corollary 17.** *Assume that there exist functions with deterministic (uniform) time complexity $2^{O(n)}$, and Π_2-circuit complexity $2^{\Omega(n)}$. If there exists a hard-on-average language in* NP *then there exists a hard-on-average problem in* TFNP.

We split the proof of Theorem 16 and Corollary 17 into two parts. First, we prove the results under the assumption that the distribution $\mathcal{D}$ has a special property we call *public-coin*, i.e., when the distribution remains hard even given the random coins used to sample from it (see Definition 8). Second, we show that this assumption does not hurt the generality of the statement, since the existence of hard distributional decision problems implies the existence of hard *public-coin* distributional decision problems.

4.1 Hardness based on public-coin distributions

We begin with the proof of the non-uniform version of our result.

Proof (Theorem 16). Fix an input size n. For simplicity of presentation (and without loss of generality), assume that to sample an instance $x \in \{0,1\}^n$ the number of random coins needed is n (otherwise, one can pad the instances to get the same effect). We begin by showing that any hard distributional decision problem $(L, \mathcal{D})$ implies the existence of a hard distributional search problem. In general, this problem will not be a total one but we will be able to use its structure to devise a total problem.

Let $\mathcal{D}$ be a hard distribution for some NP language L with associated relation R_L. The distributional search problem associated to $(L, \mathcal{D})$, i.e., given $x \leftarrow \mathcal{D}(1^n)$ to find a w such that $R_L(x, w) = 1$, is also hard. Moreover, there exists a polynomial $p(\cdot)$ such that $\Pr_{x \leftarrow \mathcal{D}}[L(x) = 1] \geq p(n)$, as otherwise, the "constant no" algorithm would contradict $\mathcal{D}$ being a hard distribution for L. For any set of $k = n^2 \cdot p(n)$ strings $s_1, \ldots, s_k \in \{0,1\}^n$, we define a distributional problem $(L_{s_1,\ldots,s_k}, \mathcal{D}')$, where the language $L_{s_1,\ldots,s_k}$ is defined by the relation

$$R_{L_{s_1,\ldots,s_k}}(r, w) = \bigvee_{i \in [k]} R_L(x_i, w), \text{ where } x_i \leftarrow \mathcal{D}(1^n; r \oplus s_i) \ ,$$

and $\mathcal{D}'$ is the uniform distribution on $\{0,1\}^n$.

First, we use the following general lemma to argue that the distributional search problem associated with $(L_{s_1,\ldots,s_k}, \mathcal{D}')$ is hard.

▶ **Lemma 18.** *Let $(L, \mathcal{D})$ be a hard distributional search problem. Let $(L', \mathcal{D}')$ be a distributional search problem related to $(L, \mathcal{D})$ that satisfies the following conditions:*

1. *L' is in an "OR" form, i.e., there exist efficiently computable functions $f_1, \ldots, f_k$ such that $R_{L'}(x', w) = \bigvee_{i \in [k]} R_L(f_i(x'), w)$ where k is some polynomially bounded function of n.*
2. *For every $i \in [k]$, the marginal distribution of $f_i(x')$ under $x' \leftarrow \mathcal{D}'$ is identical to the distribution of $x \leftarrow \mathcal{D}(1^n)$.*
3. *For any fixed instance $x^* = \mathcal{D}(1^n; r)$, the distribution $x' \leftarrow \mathcal{D}'(1^n)$ conditioned on $f_i(x') = x^*$ is efficiently sampleable (given r).*

Then $(L', \mathcal{D}')$ is a hard distributional search problem.

The proof of Lemma 18 follows by a reduction to solving the original distributional search problem $(L, \mathcal{D})$, and is deferred to Appendix A.1. Notice that $(L_{s_1,\ldots,s_k}, \mathcal{D}')$ satisfies the three conditions of Lemma 18 with respect to $(L, \mathcal{D})$: (1) the instances are in the "OR" form (where $x_i = f_i(r) = r \oplus s_i$), (2) for any $i \in [k]$ it holds that x_i is distributed as $\mathcal{D}$ since $r \oplus s_i$ is uniformly random, and (3) for any $x^* = \mathcal{D}(1^n; r^*)$, sampling from $\mathcal{D}'$ conditioned on $x_i = x^*$ can be done by setting $r = s_i \oplus r^*$. We say that $L_{s_1,\ldots,s_k}$ *is good* if it is total, i.e., if it holds that

$$\forall r \in \{0,1\}^n : L_{s_1,\ldots,s_k}(r) = 1 \ .$$

We prove that for a random choice of $s_1, \ldots, s_k$ the language $L_{s_1,\ldots,s_k}$ is good with high probability.

▶ **Lemma 19.** $\Pr_{s_1,\ldots,s_k \leftarrow \{0,1\}^{kn}}[L_{s_1,\ldots,s_k} \textit{ is good }] \geq 3/4$.

Proof. Fix any string r. If we pick s at random we get that

$$\Pr_{s \leftarrow \{0,1\}^n}[L(x) = 1 \mid x \leftarrow \mathcal{D}(1^n; r \oplus s)] \geq 1/p(n) \ .$$

This follows since for any string r the string $r \oplus s$ is uniformly random. Moreover, since for any s_i this event is independent, we get that for any fixed r (and for $k = n^2 \cdot p(n)$) it holds that:

$$\Pr_{s_1,\ldots,s_k \leftarrow \{0,1\}^{kn}}[\forall i : L(x_i) = 0 \mid x_i \leftarrow \mathcal{D}(1^n; r \oplus s_i)] \leq (1 - 1/p(n))^k \leq 2^{-n^2} \ .$$

Thus, taking a union bound over all possible $r \in \{0,1\}^n$ we get that

$$\Pr_{s_1,\ldots,s_k \leftarrow \{0,1\}^{kn}}[L_{s_1,\ldots,s_k} \text{ is not good }] \leq 2^n \cdot 2^{-n^2} \leq 1/4 \ ,$$

and thus the statement of the claim follows. ◀

Thus, for any n we fix a set of k strings non-uniformly to get a good $L_{s_1,\ldots,s_k}$, and get a language L^*, that is a hard-on-average search problem under the uniform distribution $\mathcal{D}^*$. Moreover, we get that the problem is total: for every n, every instance $r \in \{0,1\}^n$ must have a solution since we choose a good $L_{s_1,\ldots,s_k}$ for every n.[7] ◀

[7] Actually, this claim works only for large enough input sizes n. For small values of n we simply define the language to accept all instances, and thus to remain total. Notice that this does not harm the hardness of the problem.

Getting a uniform version. In the above proof of Theorem 16 we constructed a problem in non-uniform TFNP, i.e., a problem in TFNP/poly (see Definition 10). To get a uniform version of the problem (i.e., a problem in TFNP) we show how to employ derandomization techniques to find $s_1, \ldots, s_k$.

Proof (Corollary 17). Recall the definition of the language $L_{s_1,\ldots,s_k}$ from the proof of Theorem 16. Consider a circuit C that on input $s_1, \ldots, s_k$ outputs 1 if and only if $L_{s_1,\ldots,s_k}$ is good. Notice that C can be implemented by a polynomial-sized Π_2-circuit, since $s_1, \ldots, s_k$ is good if for all $r \in \{0,1\}^n$ there exists a witness (s_i, w) such that $R_L(r \oplus s_i, w) = 1$. Let G be a Nisan-Wigderson type PRG (see Definition 12) against Π_2-circuits of size $|C|$ with seed length $m = O(\log n)$. For any seed $j \in [2^m]$, let $(s_1^j, \ldots, s_k^j) = \mathsf{G}(j)$ and write $L^j = L_{s_1^j,\ldots,s_k^j}$. By Claim 19 and by pseudorandomness of G we get that

$$\Pr_{j \in [2^m]}[L^j \text{ is good}] \geq 3/4 - 1/|C| \geq 2/3 \ .$$

To derandomize the choice of $s_1, \ldots, s_k$ we define

$$L^*(r_1, \ldots, r_{2^m}) = \bigvee_{j \in [2^m]} L^j(r_j) \ ,$$

where every r_j is of length n. Accordingly, define $\mathcal{D}^*$ to sample $\bar{r} = r_1, \ldots, r_{2^m}$ uniformly at random where $r_j \in \{0,1\}^n$. Notice that $(L^*, \mathcal{D}^*)$ satisfies the following:

1. $\Pr_{\bar{r} \leftarrow \mathcal{D}^*}[L^*(\bar{r}) = 1] = 1$.
2. For any PPT A for large enough n we have: $\Pr_{\bar{r} \leftarrow \mathcal{D}^*}[L^*(\bar{r}, A(\bar{r})) = 1] \leq \mathsf{neg}(n)$.

The first item follows by the derandomization. The second item follows, since for all $j \in [2^m]$, the search problem $(L^j, \mathcal{D}')$ is hard (which follows from Lemma 18). In particular, any adversary A solving the search problem associated to $(L^*, \mathcal{D}^*)$ with noticeable probability $1/p(n)$ must solve one of he $(L^j, \mathcal{D}')$ with probability at least $1/(2^m p(n)) = 1/\mathsf{poly}(n)$, a contradiction. We get that $(L^*, \mathcal{D}^*)$ is a hard-on-average search problem, which is total, and thus in TFNP. ◀

Note that it was crucial that $\mathcal{D}$ was a public-coin distribution in order to construct $\mathcal{D}^*$ to be the uniform while maintaining hardness. This, in turn, let us define a total problem since any string is in the support of $\mathcal{D}^*$. In the next section, we show that the assumption that $\mathcal{D}$ is public-coin can be made without loss of generality.

4.2 From private coins to public coins

We prove that we can assume that our underlying distribution is public-coin without loss of generality. That is, we show that it can be (efficiently) converted to a public-coin distribution with same hardness with no additional assumptions.

▶ **Theorem 20.** *If hard-on-average* NP *languages exist then public-coin hard-on-average* NP *languages exist.*

Here we discuss two alternative versions of the proof. Though not discussed explicitly, one can see that Impagliazzo anad Levin [31] actually proved a similar statement. Their work in the context of average-case complexity showed that problems with natural hard distributions give rise to problems with *simple* hard distributions. Specifically, they showed that any problem with efficiently samplable hard distribution can be reduced to a problem with an efficiently computable hard distribution, i.e., where the cumulative distribution function is efficiently

computable. An alternative presentation of this result by Goldreich [28, Sec. 10.2.2.2] provides a step towards reducing simple distributions to hard public-coin distribution. In particular, any problem with a simple distribution can be reduced to a problem with a distribution samplable via a monotone mapping, as shown in [28, Exercise 10.14]. Given the monotonicity property, we can turn any simple distribution into a public-coin distribution. Thus, we get that decision problems with samplable distributions imply decision problems with public-coin distributions.

We provide also a self-contained proof that does not rely on the notions of samplable and simple distributions. Our approach is to use the construction of universal one-way hash functions (UOWHFs) from one-way functions [45, 48, 38]. A UOWHF is a weaker primitive than a collision resistant hash function, where an adversary chooses an input x, then it is given a hash function h sampled from the UOWHF family, and its task is to find an input $x' \neq x$ such that $h(x) = h(x')$. Therefore, any UOWHF gives rise to a hard distributional search problem (in fact, a distributional decision problem by [11]) which is public-coin: given random sample x, h find a colliding x'. We conclude the proof by showing that for any input length, a hard distribution is either already public-coin or it gives rise to a one-way function for this length. See Appendix A.2 for the complete proof.

5 Using Zero-Knowledge for TFNP Hardness

In the previous section, we have shown how to get TFNP hardness from average-case NP hardness. We either settled for a non-uniform version with no additional assumptions or used a NW-type PRG to get a uniform version. In this section we show a different approach to proving hardness of problems in TFNP. Our main technique uses Zap, a two-message witness indistinguishable proof for NP. As discussed in Section 1, the Zap can be further reduced to a single message by either fixing the first message non-uniformly or by using a NW-type PRG. In both cases, we get non-interactive witness indistinguishable proof system for NP (see Definition 11), and thus we write the proof in terms of this primitive.

The advantage of this technique is twofold. First, it is a general technique that might be used to "push" other problems inside TFNP. Second, it allows us to construct a hard problem in TFNP from injective one-way functions (see Definition 5) with at most two solutions. Formally, we prove the following theorem:

▶ **Theorem 21.** *If injective one-way functions and non-interactive witness-indistinguishable proof systems for* NP *exist, then there exists a hard-on-average problem in* TFNP *such that any instance has at most two solutions.*

Proof. We define a new total search problem and call it Invert-Either.

▶ **Definition 22** (Invert-Either)**.** Let $f\colon \{0,1\}^m \to \{0,1\}^n$ be an efficiently computable function. Let L be an NP language defined by the following relation: $R((y,y'),x) = 1$ if and only if $f(x) \in \{y,y'\}$. Let $(\mathsf{Prove},\mathsf{Verify})$ be a witness indistinguishable proof system for L. The input to the Invert-Either problem is a tuple (y,y',π), where $y,y' \in \{0,1\}^n$, and $\pi \in \{0,1\}^{\mathsf{poly}(n)}$. We ask to find a string $x \in \{0,1\}^m$ satisfying one of the following:

1. $x = 0^m$ if $\mathsf{Verify}((y,y'),\pi) = 0$.
2. $f(x) \in \{y,y'\}$ if $\mathsf{Verify}((y,y'),\pi) = 1$.

We now show that Invert-Either is a total search problem.

▶ **Lemma 23.** *The* Invert-Either *problem is in* TFNP.

Proof. Let (y, y', π) be an instance of Invert-Either. If $\mathsf{Verify}((y, y'), \pi) = 0$ then $x = 0^n$ is a solution. Otherwise if $\mathsf{Verify}((y, y'), \pi) = 1$ then, by the perfect soundness of the witness indistinguishable proof system $(\mathsf{Prove}, \mathsf{Verify})$, it follows that $(y, y') \in L$. Thus, there exists an $x \in \{0,1\}^n$ such that $f(x) = y$ or $f(x) = y'$ which is a solution for the Invert-Either instance (y, y', π). In either case there exists a solution and Invert-Either is in TFNP. ◀

We move on to show that the existence of one-way functions implies a hard on average distribution of instances of Invert-Either. Assume that there exists an efficient algorithm A that solves in polynomial time instances of Invert-Either defined relative to some one-way function f. We construct A' that inverts an image of f evaluated on a random input with noticeable probability. Given $y = f(x)$, a challenge for inverting the function f, the inverter A' proceeds as follows:

$\underline{\boldsymbol{A'(y)}}$

1. choose $x' \leftarrow \{0,1\}^n$ at random and compute $y' = f(x')$.
2. choose $b \leftarrow \{0,1\}$ at random and set $y_b = y$ and $y_{1-b} = y'$.
3. compute $\pi \leftarrow \mathsf{Prove}((y_0, y_1), x')$.
4. compute $w \leftarrow A(y_0, y_1, \pi)$.
5. output w.

Since A' computes the proof π honestly, any solution w for the Invert-Either instance (y_0, y_1, π) must be a preimage of either y or y', i.e., either $f(w) = y$ or $f(w) = y'$. If A outputs a preimage of y then A' will succeed in inverting f. However, A might output a w which is a preimage of y' which was chosen by A' and it does not help in inverting the challenge y. Our claim is that A must output a preimage of y with roughly the same probability as a preimage of y'. Formally, we show that

$$|\Pr[f(A(y_0, y_1, \pi)) = y'] - \Pr[f(A(y_0, y_1, \pi)) = y]| \leq \mathsf{neg}(n) \ .$$

It is sufficient to argue that the input tuple (y_0, y_1, π) for A produced by A' is computationally indistinguishable from an input triple produced using the actual pre-image x of the challenge y, i.e.,

$$\{y_0, y_1, \pi \leftarrow \mathsf{Prove}((y_0, y_1), x')\} \approx_c \{y_0, y_1, \pi \leftarrow \mathsf{Prove}((y_0, y_1), x)\} \ .$$

Since x and x' are chosen according to the same distribution, and y_0 and y_1 are random labels of y and y', the only way to distinguish between the two ensembles is using the proof π. However, from the witness-indistinguishability property of the proof system we get that $\mathsf{Prove}((y_0, y_1), x)$ is computationally indistinguishable from $\mathsf{Prove}((y_0, y_1), x')$ even given x and x' (and thus also given y_0 and y_1). Altogether, we get that the probability that A outputs a preimage of y is about the same probability as the probability that A outputs a preimage of y'. By our assumption, A must output either type of solution with noticeable probability. Therefore, A' succeeds in inverting the challenge y with noticeable probability. ◀

▶ Remark. We note that we get hardness of Invert-Either from any one-way function, however, the number of solutions is guaranteed to be at most two only when the one-way function is injective.

6 Open Problems

The most immediate open problem is whether it is possible to base the hardness of any of the known subclasses of TFNP on the assumption that one-way functions exist[8]. Perhaps the most plausible one is PPP: its canonical problem is given by a circuit $C\colon \{0,1\}^n \to \{0,1\}^n$ where the goal is to find a collision under C or an input x such that $C(x) = 0$. Notice that by the pigeonhole principle we get that this problem is total. The hardness of PPP was shown from one-way permutations or collision-resistant hash functions. Thus, it is a prime candidate for showing hardness from one-way functions.

Recall that Bellare and Goldwasser [10] considered the issue of self reducibility, i.e. are there languages in NP where given a decision oracle it is hard to solve *any* corresponding FNP search problem. They showed that such languages exist under the assumption that EE $\neq$ NEE (double exponential time is not equal the non-deterministic version of it). In particular the language they constructed is in P/Poly. Can standard cryptographic type assumptions and techniques be used to show such separation results?

We have shown how to use derandomization and also zero-knowledge to prove that some (hard) problems are in TFNP. Another interesting direction is to use our techniques to push into TFNP (variants of) other natural search problems which are not known to be total.

Acknowledgements. We wish to thank Oded Goldreich and Roei Tell for their invaluable help and comments.

References

1 Prabhanjan Ananth, Aayush Jain, Moni Naor, Amit Sahai, and Eylon Yogev. Universal obfuscation and witness encryption: Boosting correctness and combining security. *IACR Cryptology ePrint Archive*, 2016:281, 2016.

2 Benny Applebaum, Sergei Artemenko, Ronen Shaltiel, and Guang Yang. Incompressible functions, relative-error extractors, and the power of nondeterministic reductions. *Computational Complexity*, 25(2):349–418, 2016.

3 Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009. URL: `http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264`.

4 Sergei Artemenko, Russell Impagliazzo, Valentine Kabanets, and Ronen Shaltiel. Pseudorandomness when the odds are against you. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 9:1–9:35, 2016.

5 Gilad Asharov and Gil Segev. Limits on the power of indistinguishability obfuscation and functional encryption. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 191–209, 2015.

6 Gilad Asharov and Gil Segev. On constructing one-way permutations from indistinguishability obfuscation. In *Theory of Cryptography - 13th International Conference, TCC 2016-A, Tel Aviv, Israel, January 10-13, 2016, Proceedings, Part II*, pages 512–541, 2016.

7 Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2001. `doi:10.1007/3-540-44647-8_1`.

[8] "Provable TFNP" is a subset of TFNP containing the structured classes of it like PPP, PPAD and PLS that was defined by Goldberg and Papadimitriou [27]. It is a natural candidate for hardness proofs under as general assumption as possible.

8 Boaz Barak, Yehuda Lindell, and Salil P. Vadhan. Lower bounds for non-black-box zero knowledge. *J. Comput. Syst. Sci.*, 72(2):321–391, 2006.

9 Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. *SIAM J. Comput.*, 37(2):380–400, 2007.

10 Mihir Bellare and Shafi Goldwasser. The complexity of decision versus search. *SIAM J. Comput.*, 23(1):97–119, 1994. doi:10.1137/S0097539792228289.

11 Shai Ben-David, Benny Chor, Oded Goldreich, and Michael Luby. On the theory of average case complexity. *J. Comput. Syst. Sci.*, 44(2):193–219, 1992.

12 Nir Bitansky, Akshay Degwekar, and Vinod Vaikuntanathan. Structure vs hardness through the obfuscation lens. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:91, 2016.

13 Nir Bitansky, Omer Paneth, and Alon Rosen. On the cryptographic hardness of finding a Nash equilibrium. In *56th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, October 18-20, 2015*, pages 1480–1498, 2015.

14 Nir Bitansky and Vinod Vaikuntanathan. A note on perfect correctness by derandomization. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:187, 2015.

15 Manuel Blum and Sampath Kannan. Designing programs that check their work. *J. ACM*, 42(1):269–291, 1995.

16 Gilles Brassard. Relativized cryptography. *IEEE Trans. Information Theory*, 29(6):877–893, 1983.

17 Harry Buhrman, Lance Fortnow, Michal Koucký, John D. Rogers, and Nikolai K. Vereshchagin. Does the polynomial hierarchy collapse if onto functions are invertible? *Theory Comput. Syst.*, 46(1):143–156, 2010.

18 Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player Nash equilibria. *J. ACM*, 56(3), 2009.

19 Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM J. Comput.*, 39(1):195–259, 2009.

20 Constantinos Daskalakis and Christos H. Papadimitriou. Continuous local search. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 790–804, 2011.

21 Cynthia Dwork and Moni Naor. Zaps and their applications. *SIAM J. Comput.*, 36(6):1513–1543, 2007.

22 Cynthia Dwork, Moni Naor, and Omer Reingold. Immunizing encryption schemes from decryption errors. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pages 342–360, 2004.

23 Shimon Even, Alan L. Selman, and Yacov Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61(2):159–173, 1984.

24 Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 416–426, 1990.

25 Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, pages 40–49, 2013. doi:10.1109/FOCS.2013.13.

26 Sanjam Garg, Omkant Pandey, and Akshayaram Srinivasan. Revisiting the cryptographic hardness of finding a Nash equilibrium. In *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, pages 579–604, 2016.

27 Paul W. Goldberg and Christos H. Papadimitriou. Towards a unified complexity theory of total functions, 2016. Unpublished manuscript. URL: http://www.cs.ox.ac.uk/people/paul.goldberg/papers/paper-2.pdf.

28 Oded Goldreich. *Computational complexity - a conceptual perspective*. Cambridge University Press, 2008.

29 Pavel Hubáček and Eylon Yogev. Hardness of continuous local search: Query complexity and cryptographic lower bounds. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:63, 2016.

30 Russell Impagliazzo. A personal view of average-case complexity. In *Structure in Complexity Theory Conference*, pages 134–147. IEEE Computer Society, 1995. doi:10.1109/SCT.1995.514853.

31 Russell Impagliazzo and Leonid A. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume II*, pages 812–821, 1990.

32 Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *FOCS*, pages 230–235. IEEE Computer Society, 1989. doi:10.1109/SFCS.1989.63483.

33 Russell Impagliazzo and Moni Naor. Decision trees and downward closures. In *Proceedings: Third Annual Structure in Complexity Theory Conference, Georgetown University, Washington, D. C., USA, June 14-17, 1988*, pages 29–38, 1988.

34 Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 220–229, 1997.

35 Emil Jeřábek. Integer factoring and modular square roots. *J. Comput. Syst. Sci.*, 82(2):380–394, 2016.

36 David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *J. Comput. Syst. Sci.*, 37(1):79–100, 1988. doi:10.1016/0022-0000(88)90046-3.

37 Jeff Kahn, Michael E. Saks, and Clifford D. Smyth. The dual BKR inequality and Rudich's conjecture. *Combinatorics, Probability & Computing*, 20(2):257–266, 2011.

38 Jonathan Katz and Chiu-Yuen Koo. On constructing universal one-way hash functions from arbitrary one-way functions. *IACR Cryptology ePrint Archive*, 2005:328, 2005. URL: http://eprint.iacr.org/2005/328.

39 Ilan Komargodski, Tal Moran, Moni Naor, Rafael Pass, Alon Rosen, and Eylon Yogev. One-way functions and (im)perfect obfuscation. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 374–383, 2014.

40 Clemens Lautemann. BPP and the polynomial hierarchy. *Inf. Process. Lett.*, 17(4):215–217, 1983.

41 László Lovász, Moni Naor, Ilan Newman, and Avi Wigderson. Search problems in the decision tree model. *SIAM J. Discrete Math.*, 8(1):119–132, 1995.

42 Mohammad Mahmoody and David Xiao. On the power of randomized reductions and the checkability of SAT. In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity, CCC 2010, Cambridge, Massachusetts, June 9-12, 2010*, pages 64–75, 2010.

43 Nimrod Megiddo and Christos H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theor. Comput. Sci.*, 81(2):317–324, 1991. doi:10.1016/0304-3975(91)90200-L.

44 Moni Naor. Bit commitment using pseudorandomness. *Journal of Cryptology*, 4(2):151–158, 1991. doi:10.1007/BF00196774.

45 Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 33–43, 1989.

46 Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.

47 Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.*, 48(3):498–532, 1994. doi:10.1016/S0022-0000(05)80063-7.

48 John Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 387–394, 1990.

49 Alon Rosen, Gil Segev, and Ido Shahaf. Can PPAD hardness be based on standard cryptographic assumptions? *Electronic Colloquium on Computational Complexity (ECCC)*, 23:59, 2016.

50 Steven Rudich. *Limits on the Provable Consequences of One-way Functions*. PhD thesis, University of California at Berkeley, 1989.

51 Amit Sahai and Brent Waters. How to use indistinguishability obfuscation: deniable encryption, and more. In *STOC*, pages 475–484. ACM, 2014. doi:10.1145/2591796.2591825.

52 Daniel R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding*, pages 334–345, 1998.

A Detailed Proofs

A.1 Proof of Lemma 18

We give the full proof of Lemma 18 that we use in the proof of Section 4.1. The proof follows by a reduction to the original distributional problem $(L, \mathcal{D})$. Given a challenge x^* for $(L, \mathcal{D})$ we can create an instance σ of the search problem associated with $(L', \mathcal{D}')$ such that the challenge x^* is embedded in a random location among the "ORed" instances in σ. This can be done efficiently by the first and the third item. Then, by the second item we get that if $x^* \in L$ then a solution for σ will contain a solution for x^* with probability at least $1/k$. Overall, we gain a polynomial advantage for solving x^* which contradicts the hardness of $(L, \mathcal{D})$.

▶ Lemma 18 (restated). *Let $(L, \mathcal{D})$ be a hard distributional search problem. Let $(L', \mathcal{D}')$ be a distributional search problem related to $(L, \mathcal{D})$ that satisfies the following conditions:*

1. *L' is in an "OR" form, i.e., there exist efficiently computable functions $f_1, \ldots, f_k$ such that $R_{L'}(x', w) = \bigvee_{i \in [k]} R_L(f_i(x'), w)$ where k is some polynomially bounded function of n.*
2. *For every $i \in [k]$, the marginal distribution of $f_i(x')$ under $x' \leftarrow \mathcal{D}'$ is identical to the distribution of $x \leftarrow \mathcal{D}(1^n)$.*
3. *For any fixed instance $x^* = \mathcal{D}(1^n; r)$, the distribution $x' \leftarrow \mathcal{D}'(1^n)$ conditioned on $f_i(x') = x^*$ is efficiently sampleable (given r).*

Then $(L', \mathcal{D}')$ is hard distributional search problem.

Proof. Assume towards contradiction that there exist an adversary A and a polynomial $p(\cdot)$ such that

$$\Pr_{x \leftarrow D'(1^n; r)}[R_{L'}(x, A(x, r)) = 1] \geq 1/p(n) \ .$$

Then, we construct an adversary A' that solves the search problem $(L, \mathcal{D})$.

$\underline{A'(x)}$

1. Choose $i \in [k]$ at random.
2. Sample $x' \leftarrow \mathcal{D}'$ conditioned on $f_i(x') = x$ (this can be performed efficiently due to item 3.).
3. Output $w \leftarrow A(x')$.

Notice that since x is sampled from $\mathcal{D}$ and the marginal distribution of $f_i(x')$ is identical to that of $\mathcal{D}$ (item 2.), we get that x' is distributed exactly as a sample from $\mathcal{D}'$. Therefore, when computing w we have that A sees exactly the distribution it expects, and it will find a valid solution to x' with probability at least $1/p(n)$.

If $x \in L$, then with probability $1/p(n)$ we have that A will give us a solution to x' and since x' is in the "OR" form (item 1.), with probability $1/k$ that solution will solve x (and otherwise it answers at random). Thus, the probability for A' to solve x is at least $1/(k \cdot p(n))$ which contradicts $(L, \mathcal{D})$ being a hard distributional search problem. ◀

A.2 Proof of Theorem 20

We give the proof of the following theorem establishing that private-coin distributional decision problems imply existence of public-coin distributional decision problems.

▶ Theorem 20 (restated). *If hard-on-average* NP *languages exist then public-coin hard-on-average* NP *languages exist.*

Proof. We begin by showing that if one-way functions exist then there are hard-on-average distributions that are public-coin. This part of the proof follows by combining known transformations. First, it is known that if one-way functions exist then universal one-way hash functions (UOWHFs) exist [45, 48, 38]. A UOWHF is a family of compressing functions $\mathcal{H}$ that have the following security requirement: for a random x and $h \in \mathcal{H}$ it is hard for any PPT algorithm to find an $x \neq x'$ such that $h(x) = h(x')$. We note that the constructions of such families are in fact public-coin: to sample h no private coins are used. Thus, we can define the following search problem: given x, h find an appropriate collision x'.[9] Second, we can apply to this search problem a transformation of Ben-David et al. [11] for converting any average-case search problem into an average-case decision problem. Although it was not mentioned explicitly in their work, we observe when applied to a search problem that has a public-coin distribution the resulting decision problem is public-coin as well.

We have shown how to get a hard public-coin distributional problem from one-way function. We want to show how to get the same result from any hard distributional problem. Let $\mathcal{D}$ be a hard-on-average NP distribution for some language L. If $\mathcal{D}$ is public-coin, then we are done. Assume that $\mathcal{D}$ is not public-coin.

If we were able to prove a statement of the form "private-coin distributions imply one-way functions", then by applying the above two transformations we would be done. But what exactly is a "private-coin" distribution? Our only assumption is that $\mathcal{D}$ is not public-coin. It might be the case that for some (infinitely many) input sizes the distribution is public-coin and for some (infinitely many) it is not. Thus, the function that we get will be hard to invert only on the input sizes that the distribution is not public-coin. Then, the distribution that we get from this function will be public-coin only for the same input sizes. However, for the

[9] Notice that this search problem problem is not in TFNP since no such collision x' might exist.

rest of the input sizes, the distribution was already public-coin! Thus, by combining the two we can get a hard public-coin distribution for any input size.

One subtle issue is that we do not necessarily know for which input sizes is the distribution public-coin and for which not. Thus, for any input size n we apply both methods: we two samples using randomness r_1 and r_2. For r_1 we release $r_1, \mathcal{D}(1^n, r_1)$, and we r_2 to sample from the distribution constructed from the one-way function, as described above. Finally, we take the "AND" of the both. For any n we know that one of the two will be hard, and thus overall we get hardness for any input size.

We are left to show how to construct one-way functions that are hard to invert for the non-public input sizes of the distribution. Assume that $\mathcal{D}$ is not public-coin. Then, there exist an efficient algorithm A, and a polynomial p such that for infinitely many input sizes it holds that:

$$\Pr_{r,A}[A(r,x) = L(x) : x \leftarrow D_n(r)] \geq 1/2 + 1/p(n) \ . \tag{A.1}$$

Let $\mathcal{N}$ be the infinite set of $n \in \mathbb{N}$ such that (A.1) holds. We define the function family $f = \{f_n(r) = D_n(r)\}_{n\in\mathbb{N}}$. We claim that it is infeasible to invert f for the input sizes in $\mathcal{N}$:

▶ **Lemma 24.** *For any PPT A there exists a negligible function $\mathsf{neg}(\cdot)$ such that for all $n \in \mathcal{N}$:*

$$\Pr_{r\in\{0,1\}^n}[A(f_n(r)) \in f_n^{-1}(f_n(r))] \leq \mathsf{neg}(n) \ .$$

Proof. Impagliazzo and Luby [32] showed that if one-way functions do not exist, then it is not only possible to invert a function f, but also to get a close to uniform inverse. Formally, if A is an adversary that inverts f then there exists a constant $c > 0$ such that A outputs a distribution that is $1/n^c$-close to a uniform distribution over the inverses.

Thus, suppose that f is not one-way as stated in the Lemma. Then, given $y = f(r)$ we can run the inverter on y and get r' such that $f(r) = f(r')$ with probability $1/p(n)$ for some polynomial p. Moreover, there exists a constant c such that the distribution of r' is $1/n^c$ close to a uniform one. The high-level idea is that if we run $A(r', x)$ then we get the correct answer with high probability, thus we are able to decide L relative to $\mathcal{D}$ with high probability.

Formally, we verify that $\mathcal{D}(r) = \mathcal{D}(r')$. If this is not the case then we answer randomly. Assume that $\mathcal{D}(r) = \mathcal{D}(r')$. Let

$$\mathcal{R}_x = \{r : \Pr_A[A(r,x) = L(x)] \geq 1/2 + 1/p(n)\}.$$

We say that x is good if $\Pr_r[r \in \mathcal{R}_x] \geq 1/2$. By (A.1) we get that the probability that x is good is at least half. If x is good, then we get that $A(r', x) = L(x)$ with probability at least $1/2 + 1/p(n) - 1/n^c$. Altogether, we get a polynomial advantage above $1/2$ in deciding L. ◀

The above Lemma concludes the proof. ◀